

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

ANÁLISE DE UM ALGORITMO DE FLUXO EM
REDES COM REMOÇÃO DE ARCOS

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CA
TARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA

NELCY DABROWSKI DE ARAÚJO

FLORIANÓPOLIS
SANTA CATARINA - BRASIL
SETEMBRO - 1981


ANÁLISE DE UM ALGORITMO DE FLUXO EM
REDES COM REMOÇÃO DE ARCOS

NELCY DABROWSKI DE ARAÚJO

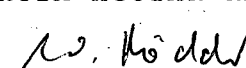
ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE
"MESTRE EM ENGENHARIA"
ESPECIALIDADE ENGENHARIA DE PRODUÇÃO E APROVADA EM SUA
FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO.


PROF. JOHN ROBERT MACKNESS
COORDENADOR

APRESENTADA PERANTE A BANCA EXAMINADORA COMPOSTA DOS PROFESSORES:


PROF. PAULO RENÉCIO NASCIMENTO, M.Sc.
PRESIDENTE


PROFA. MARCIA AGUIAR RABUSKE, D.Sc.


PROF. WILHELM RÖDDER, Ph.D.



UFSC-BU

Aos meus pais

Adão e

Edviges.

A G R A D E C I M E N T O S .

Pela excelente, orientação e pelo estímulo na execução deste trabalho, meu agradecimento sincero ao professor PAULO R. NASCIMENTO.

Aos colegas ROSITA S.V. TRIDAPALLI, VÂNIA C. TAVARES e FREDERICO A. ALVAREZ agradeço pelo incentivo, à RITA de C.B. NASCIMENTO pela datilografia.

À amiga EVANILDA TEIXEIRA sou grata pela colaboração.

R E S U M O

O presente trabalho trata do problema de remoção de arcos em uma rede de fluxo, tal que a redução no fluxo máximo entre a fonte e o sumidouro é maximizada.

Inicialmente é feito um estudo de caminho mais curto em grafos e fluxo em redes.

Segue-se uma análise do algoritmo de Wollmer, que é dividida em duas partes. A primeira, para a obtenção do valor do caminho mais curto, e a segunda para obter esse caminho e os arcos que falharam. Depois de vários testes, a análise foi concluída com a observação de que esse é um processo válido para problemas particulares.

Propõe-se, então, um algoritmo genérico para fazer a análise de sensibilidade em uma rede. Na primeira parte utiliza-se o algoritmo em estudo, com algumas modificações; na segunda aplica-se a técnica de apontadores. Um exemplo ilustra o processo.

A B S T R A C T

This dissertation analyses the problem of removing arcs in a network such that the reduction in the maximum flow between an origin and destination is maximised.

Initially a study of critical path methods is made. This is followed by an analysis of the Wollmer Algorithm which is divided in two parts. The first part is used to obtain the flow value of the critical path and the second part to obtain the set of arcs which make up the critical path and those which are removed. The analysis shows that this algorithm works only for specific problems.

The dissertation proposes a general algorithm for sensitivity analysis of a network. The first part, which some modifications, is equivalent to the Wollmer Algorithm; in the second part the use of the critical path indicators is made to speed up the computation process.

An example illustrates the use of this algorithm.

S U M Á R I O

	Pag.
 CAPÍTULO I	
1. INTRODUÇÃO	1
1.1. O Estudo de um Algoritmo de Fluxo em Redes	
com Remoção de Arcos	2
1.1.1. Objetivo	2
1.1.2. Importância	2
1.1.3. Limitações	3
1.1.4. Metodologia desenvolvida	4
1.2. Estrutura do Trabalho	4
 CAPÍTULO II	
2. CAMINHOS E FLUXOS	6
2.1. Definições	6
2.2. Caminho mais Curto em Grafos	9
2.3. A Rede Dual	12
2.4. O Problema de Caminho mais Curto com Remoção	
de Arcos	13
 CAPÍTULO III	
3. ANÁLISE DO ALGORITMO DE WOLLMER	14
3.1. Primeira Parte do Algoritmo de Wollmer: Determi-	
nação do Valor do Caminho mais Curto	14
3.1.1. Fluxograma da Primeira Parte do Algoritmo	
de Wollmer	15

3.1.2. Exemplos do Algoritmo Original	16
3.2. Modificações	20
3.2.1. Exemplo do Algoritmo com Modificações	23
3.3. Comparação do Algoritmo de Wollmer com o Algoritmo Modificado	25
3.4. Segunda Parte do Algoritmo de Wollmer: Determi- nação do Caminho mais Curto	36
3.4.1. Observações Relativas à Segunda Parte do Algoritmo	44
CAPÍTULO IV	
4. ALGORITMO PROPOSTO - CAMI	45
4.1. Fluxograma do Algoritmo CAMI para obtenção do Valor do Caminho mais Curto	49
4.2. Exemplo de CAMI	49
CAPÍTULO V	
5. SUGESTÕES PARA NOVAS PESQUISAS	53
BIBLIOGRAFIA	54

C A P Í T U L O I

1. INTRODUÇÃO

A teoria dos grafos foi desenvolvida para suprir a necessidade da resolução de problemas práticos de várias especialidades, tendo estruturas e propriedades comuns. A teoria trata das relações existentes entre os elementos de um ou mais conjuntos, não se preocupando com o conteúdo desses elementos¹.

São muitos os problemas práticos que exigem esse tratamento, como aqueles envolvendo Pesquisa Operacional, transporte, química, organização, economia, sociologia, etc.

Atualmente tem sido importante a formalização desses conceitos em ciência de computação. Por isso vêm-se desenvolvendo técnicas cada vez mais eficientes para processar em computador problemas que envolvam grafos, sendo que os que tratam do fluxo em redes são considerados os mais importantes.

Em geral, quando não for necessário fazer distinção, será utilizado o termo "rede" tanto para uma rede como para o grafo que lhe serve de suporte. Por exemplo, um mapa rodoviário define um grafo. O mesmo mapa é considerado uma rede se sobre ele são colocadas as distâncias das diferentes posições.

Pelo grande número de aplicações a problemas bastante diversos o desenvolvimento da teoria gerou problemas de notação, de nomenclatura, e até quanto a importância de determina-

¹FURTADO, A.L. Teoria dos... p.ix.

dos tópicos. Este desenvolvimento é motivado por duas considerações principais. No plano prático, problemas de otimização, importantes por suas aplicações concretas. Por outro lado, existem problemas onde o conceito de rede não aparece diretamente, mas cuja solução passa pela pesquisa de um caminho ou de um fluxo em uma rede associada (por exemplo, rede de investimentos).

1.1. O Estudo de um Algoritmo de Fluxo em Redes com Remoção de Arcos

1.1.1. Objetivo

O objetivo foi obter um algoritmo que fosse eficiente para a solução de problemas genéricos de redes com remoção de arcos, e computacionalmente viável em termos de tempo de execução.

Este objetivo surgiu quando da análise do algoritmo de Wollmer² que não apresentou os resultados esperados para certos problemas.

1.1.2. Importância

Quando se deseja executar o processo de obtenção do caminho mais curto em redes, não considerando a remoção de arcos, existem vários algoritmos que podem especificar até determinadas características particulares do problema. Porém, se o pro-

²WOLLMER, R. Removing Arcs... p. 934-40.

blema for de uma rede em que se queira verificar qual a influência que sofrerá quando um ou mais arcos forem suprimidos, não há muitas opções.

Neste trabalho foi feita uma análise crítica do algoritmo de Wollmer, cuja referência é "WOLLMER, R. Removing Arcs from a Network. Operations Research, 12:934-40, 1964", que será citado durante todo o trabalho. Mostrou-se que não apresenta resultados satisfatórios para algumas redes sendo, então, proposto um algoritmo que pretende resolver o problema formulado com um pequeno tempo computacional. Uma rede de 50 nós e 71 arcos levou menos de um segundo de CPU em um computador IBM 4341.

Este algoritmo será de grande valia para que vários tipos de decisões possam ser tomadas. Por exemplo: em termos de preparação da rede para ocorrência de reduções planejadas (redes de transporte), ou tratando de custos, a sensibilidade da rede (redes de investimento).

1.1.3. Limitações

O processo de determinação de falhas nos arcos no problema primal considera que os mesmos tenham seus comprimentos reduzidos a zero no problema dual; não foi estudada a possibilidade de falhas parciais.

Uma restrição na aplicação do processo é que a rede seja planar fonte-sumidouro.

1.1.4. Metodologia Desenvolvida

Na pesquisa inicial sobre o assunto tratado procurou-se trabalhos com objetivos semelhantes para que se pudesse fazer interpretações e até mesmo comparações de processos. Estes não foram encontrados.

Iniciou-se, então, a fase de programação do algoritmo de Wollmer na qual apareceram determinados problemas que exigiram bastante atenção. Alguns deles foram resolvidos pelas próprias técnicas de programação. Outros, chegou-se à conclusão, eram defeitos do próprio algoritmo. O passo seguinte foi fazer modificações, procurando-se não alterar completamente a estrutura deste algoritmo, como ocorreu na primeira parte. Já na segunda parte foi maior a conveniência de não utilizar o algoritmo original. Neste algoritmo, depois de calculado o valor do caminho mais curto, era necessário refazer todos os cálculos para obter o caminho percorrido, parecendo, por isso, ser computacionalmente ineficiente, mesmo que fosse corrigido.

1.2. Estrutura do Trabalho

Algumas definições e considerações sobre o caminho mais curto em grafos, o problema de fluxo em redes e a obtenção da rede dual são feitas no capítulo 2. Este contém ainda a especificação do problema a ser tratado, que é o de caminho mais curto em redes com arcos sujeitos a falhas, que relaciona todos os tópicos apresentados.

Uma análise do algoritmo de Wollmer é feita no capítulo 3, dando exemplos, mostrando seus defeitos e fazendo modificações. Ainda neste capítulo é apresentada uma comparação entre o algoritmo original e o modificado.

O capítulo 4 contém um algoritmo que propõe resolver o problema formulado, a partir das modificações já apresentadas no capítulo anterior. Este algoritmo ao procurar o valor do caminho mais curto guarda em um vetor qual é o caminho percorrido, que está relacionado ao número permitido de falhas.

Algumas sugestões para novas pesquisas são apresentadas no capítulo 5.

C A P Í T U L O I I

2. CAMINHOS E FLUXOS

A uma rede na qual se considera a circulação de fluxo, associa-se um grafo; por analogia, esse grafo é comumente chamado rede. Em Pesquisa Operacional os problemas de circulação de fluxo são considerados pela maioria dos autores como sendo os mais importantes da teoria dos grafos, pelo número de aplicações existentes e pela diversidade de contextos nos quais eles são encontrados.

Os problemas de fluxo podem ser:

- o do movimento em uma rede existente, na qual se deseja uma política de circulação; por exemplo, para um problema de transporte, no qual a rede é formada por pontos de abastecimento e o fluxo é representado por veículos. Deseja-se encontrar qual o melhor itinerário a ser percorrido, passando por todos os pontos de abastecimento;

- o de construção ou modificação de uma rede, na qual se deseja que o fluxo possa ser especificado. Por exemplo, para redes telefônicas, em primeiro lugar tem-se o problema de implantação de um sistema telefônico em uma cidade, e em segundo, o problema de ampliação de um sistema telefônico já existente.

2.1. Definições

. Um grafo G é definido por um par (X,U) , onde: X é um conjunto $\{x_1, x_2, \dots, x_n\}$ de elementos chamados vértices ou

nós, e U é um conjunto (u_1, u_2, \dots, u_m) de elementos chamados arcos, ligando todos ou alguns destes nós.

. Para um arco $u=(x,y)$, o vértice x é chamado extremidade inicial e o vértice y é chamado extremidade terminal.

. Um vértice y é chamado um sucessor do vértice x se há um arco com x como sua extremidade inicial e y como sua extremidade terminal. Um vértice y é chamado antecessor do vértice x se há um arco da forma (y,x) .

. Uma cadeia é uma seqüência $u=(u_1, u_2, \dots, u_q)$ de arcos de G tal que cada arco da seqüência tem uma extremidade em comum com seu antecessor na seqüência e sua outra extremidade em comum com seu sucessor na seqüência. O número de arcos da seqüência é o comprimento (ou valor) da cadeia u .

. Um caminho de comprimento (ou valor) q é uma cadeia $u=(u_1, u_2, \dots, u_i, \dots, u_q)$ na qual a extremidade terminal do arco u_i é a extremidade inicial do arco u_{i+1} para todo $i < q$.

. Um ciclo é uma cadeia tal que:

- 1) nenhum arco aparece duas vezes na seqüência, e
- 2) as duas extremidades da cadeia estão no mesmo vértice.

. Um circuito é um ciclo $u=(u_1, u_2, \dots, u_q)$ tal que para todo $i < q$ a extremidade terminal de u_i é a extremidade inicial de u_{i+1} .

. Um grafo conexo é um grafo que contém uma cadeia $u[x,y]$ para cada par (x,y) de vértices distintos.

. Um grafo G é dito ser planar se é possível repre

sentá-lo em um plano no qual os vértices são pontos distintos, os arcos são curvas simples e quaisquer dois arcos não se cruzam. A representação de G que satisfaz os requisitos acima, em um plano, é chamado grafo planar topológico. (Durante todo o trabalho o termo grafo planar indicará tanto o próprio grafo, como sua representação).

. Uma face de G é definida como uma região do plano limitada por arcos tais que quaisquer dois pontos em uma região podem ser ligados por uma curva contínua que não contém arcos ou vértices.

. Considere-se um grafo que a cada arco (x,y) tem associado um número inteiro não negativo b_{xy} , chamado sua capacidade, que representa a quantidade de fluxo que pode passar através de cada arco. Um fluxo é um conjunto de números inteiros não negativos f_{xy} se satisfaz às seguintes restrições:

$$1. \sum_x f_{xy} - \sum_k f_{yk} = \begin{cases} -v & \text{se } y = s \\ 0 & \text{se } y \neq s, t \\ v & \text{se } y = t \end{cases}$$

2. $0 \leq f_{xy} \leq b_{xy}$, sendo s e t dois nós especiais do grafo, chamados fonte e sumidouro, respectivamente, e v um valor não negativo chamado valor do fluxo.

A primeira restrição expressa o fato de que o fluxo que entra em um nó é igual ao fluxo que sai deste nó, exceto no nó fonte e no nó sumidouro. A segunda restrição mostra que o fluxo é sempre limitado pela capacidade do arco.

. Considera-se uma rede como um grafo com um fluxo de algum tipo em seus arcos.

. Uma rede é planar fonte-sumidouro quando continua planar, se um arco conectando fonte e sumidouro é adicionado a uma rede planar.

. Um corte é o conjunto dos arcos que ligam um nó em A a um nó em B, sendo A e B partições de nós, tal que a fonte está em A e o sumidouro em B. O valor do corte é a soma das capacidades desses arcos. Um corte separando a fonte do sumidouro com valor mínimo é chamado corte mínimo.

. Teorema (Ford e Fulkerson³): Fluxo máximo-corte mínimo. Para qualquer rede o valor do fluxo máximo da fonte para o sumidouro é igual à capacidade de um corte mínimo separando a fonte e o sumidouro.

2.2. Caminho mais Curto em Grafos

Dos problemas de caminhos em grafos o problema de caminho mais curto é considerado o mais importante por sua vasta aplicação; por exemplo, em redes de telecomunicações, associadas as distâncias físicas, ou para outro tipo de rede, caminhos constituídos por etapas decisórias.

Na procura do caminho mais curto pode-se considerar vários itens como:

- o interesse na obtenção de caminhos de um vértice dado aos vértices restantes, ou entre todos os pares de vértices;
- a existência ou não de circuitos;

³FORD, L.K. e FULKERSON, D.R. Flows in ... p.10-14.

- a existência ou não, de restrições relativas ao valor do caminho ou ao número de arcos;
- a necessidade de se determinar o caminho ou apenas seu valor;
- o problema é repetitivo ou basta resolvê-lo uma vez.

Essa grande diversidade de situações deu origem a um grande número de algoritmos que podem ser classificados por tipos, de acordo com as técnicas de busca utilizadas.

Boaventura Netto⁴ faz a classificação dos algoritmos em duas classes:

- por ajustes sucessivos: seja $\ell(u_{ki})$ o comprimento do menor caminho entre x_k e x_i ; e λ_{ki} uma estimativa de $\ell(u_{ki})$. Como o valor correto de $\ell(u_{ki})$ é desconhecido, este valor é substituído por λ_{ki} . Procura-se melhorar essa estimativa considerando os caminhos que passam pelos vértices x_j que sejam antecessores de x_i : $\lambda_{ki}^+ = \min_j |\lambda_{ki}, \lambda_{kj} + d_{ji}|$. Essa expressão é aplicada sucessivamente aos vértices até que não seja possível reajustar nenhum valor de comprimento, quando terão sido atingidos os comprimentos dos caminhos mínimos;

- por indução e ajuste: operam com subgrafos progressivamente maiores, até que se tenha abrangido todo o grafo; os valores vão se tornando definitivos e quando isso ocorrer com todos os vértices se procura o itinerário correspondente ao valor mínimo achado, voltando para trás.

⁴ BOAVENTURA NETTO, P.O. Teoria e Modelos... p.162-164.

Distingue-se ainda os algoritmos destinados a encontrar os caminhos mínimos a partir de um vértice dado e os matriciais, que permitem determinar os caminhos mínimos entre todos os pares de vértices.

Segundo Boaventura Netto⁵, os algoritmos mais conhecidos, dados pelos nomes de seus autores são:

1. Ajustes sucessivos:

- um vértice fixo:

Ford

Bellman-Kalaba

Roy

Moore

Dantzig

Yen

Dijkstra

- todos os pares de vértices:

Bellman-Kalaba generalizado

Farbey-Land-Murchland

Hu

2. Indução e ajuste:

- um vértice fixo:

Dantzig-Blattner-Rao

- todos os pares de vértices:

Dantzig

⁵BOAVENTURA NETTO, P.O., op.cit..

Floyd

Shimbel

Yen

2.3. A Rede Dual

A rede dual de uma rede, chamada primal, quando de finida, é outra rede na qual os arcos em vez de terem capacidades, tem comprimentos. Uma rede dual é definida se e somente se a rede primal é planar fonte-sumidouro.

A construção de uma rede dual quando definida pode ser feita da seguinte forma:

a) Coloque um nó em cada face da rede primal. A fonte será o nó na face finita, que é definida pelo arco artificial (ligando fonte e sumidouro) e o sumidouro será o nó na face infinita, limitada pelos arcos externos.

b) Para cada arco na rede primal (exceto o arco ar tificial) construa um arco na rede dual que o intercepte e ligue os nós das faces vizinhas.

c) Atribua a cada arco da rede dual um comprimento igual a capacidade do arco primal que ele corta.

Repetindo a operação com a rede dual obtida, encon tra-se novamente a rede primal, uma propriedade dos pares primal-dual, quando a rede primal é conexa.

Há uma correspondência biunívoca entre os cortes da rede primal e as rotas da rede dual, sendo que uma rota atra vés da rede dual é qualquer caminho de sua fonte para seu sumidouu

ro. Assim, o problema de encontrar o corte mínimo pode ser transformado em um problema de encontrar o caminho mais curto.

2.4. O Problema de Caminho mais Curto com Remoção de Arcos

Para qualquer rede o valor do fluxo máximo é igual a capacidade de um corte mínimo separando a fonte e o sumidouro (teorema do fluxo máximo - corte mínimo), que é igual ao valor do caminho mais curto através da rede dual. Para resolver o problema de caminho mais curto quando não há redução na capacidade de nenhum arco, podem ser utilizados vários algoritmos, como por exemplo os citados no item 2.2.

Neste trabalho será tratado o problema de remover n arcos em uma rede, de forma que a redução no fluxo máximo entre os nós fonte e sumidouro seja maximizada. É especificado inicialmente o número máximo e não quais os arcos que sofrerão redução. Agora, o problema de determinar qual o valor do fluxo máximo que passa pela rede com a retirada de n arcos, é equivalente a obter o valor do caminho mais curto na rede dual, com a retirada de n arcos.

3. ANÁLISE DO ALGORITMO DE WOLLMER

Procura-se aqui determinar qual o valor do fluxo máximo que passa por uma rede, com a retirada de n arcos. Isto é feito através da obtenção do valor do caminho mais curto na rede dual, com a retirada de n arcos. Determina-se o comprimento do caminho da fonte s até o nó a com os comprimentos de i ou menos de seus arcos reduzidos a zero, $D_{a,i}$; até que se obtenha o menor comprimento, $L_{a,i}$. Para $i=0,1,\dots,n$, tem-se $D_{a,0}; D_{a,1}; \dots; D_{a,n}$, tal que inicialmente $D_{a,i} \geq L_{a,i}$. Calcula-se até que todos os $D_{a,i} = L_{a,i}$. Assim, $D_{t,n}$ será o comprimento do caminho desejado.

3.1. Primeira Parte do Algoritmo de Wollmer: Determinação do Valor do Caminho mais Curto

Na apresentação do algoritmo de Wollmer e nas modificações efetuadas, procurou-se utilizar a nomenclatura original, para mostrar a dificuldade de interpretação e evitar problemas dessa natureza.

Seja: s - fonte

t - sumidouro

$\ell(a,b)$ - comprimento do arco (a,b)

O algoritmo é constituído dos seguintes passos:

1. Para $i=0,1,\dots,n$ faça $D_{s,i}=0$ e $D_{a,i}=\infty$ (para $a \neq s$)

Seja $k=0$.

2. Para cada arco (a,b) , verifique:

a) Se $D_{a,k} > D_{b,k} + \ell(a,b)$ faça $D_{a,k} = D_{b,k} + \ell(a,b)$

b) Se $D_{b,k} > D_{a,k} + \ell(a,b)$ faça $D_{b,k} = D_{a,k} + \ell(a,b)$

c) Se $k \geq 1$ e $D_{a,k} > D_{b,k-1}$ faça $D_{a,k} = D_{b,k-1}$

d) Se $k \geq 1$ e $D_{b,k} > D_{a,k-1}$ faça $D_{b,k} = D_{a,k-1}$

Até que nenhuma mudança possa ser feita.

3. Então:

a) Se $k < n$ aumente k de 1 e vá a 2.

b) Se $k = n$ termine, com $D_{t,n}$ sendo o valor desejado.

3.1.1. Fluxograma da Primeira Parte do Algoritmo de Wollmer

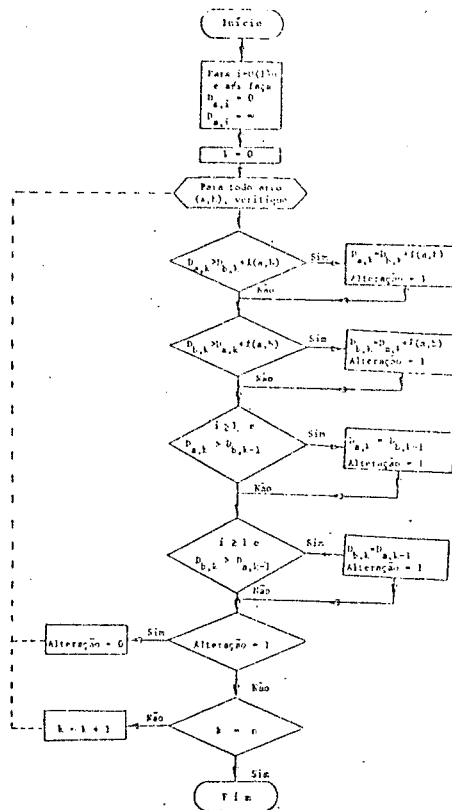


FIGURA 1 - Primeira parte do algoritmo de Wollmer.

3.1.2. Exemplos do algoritmo original

Exemplo 1: A figura 2 mostra uma rede onde as capacidades dos arcos da rede primal (ou os comprimentos dos arcos na rede dual) estão indicados. A rede primal está indicado com linha contínua e a rede dual, tracejada.

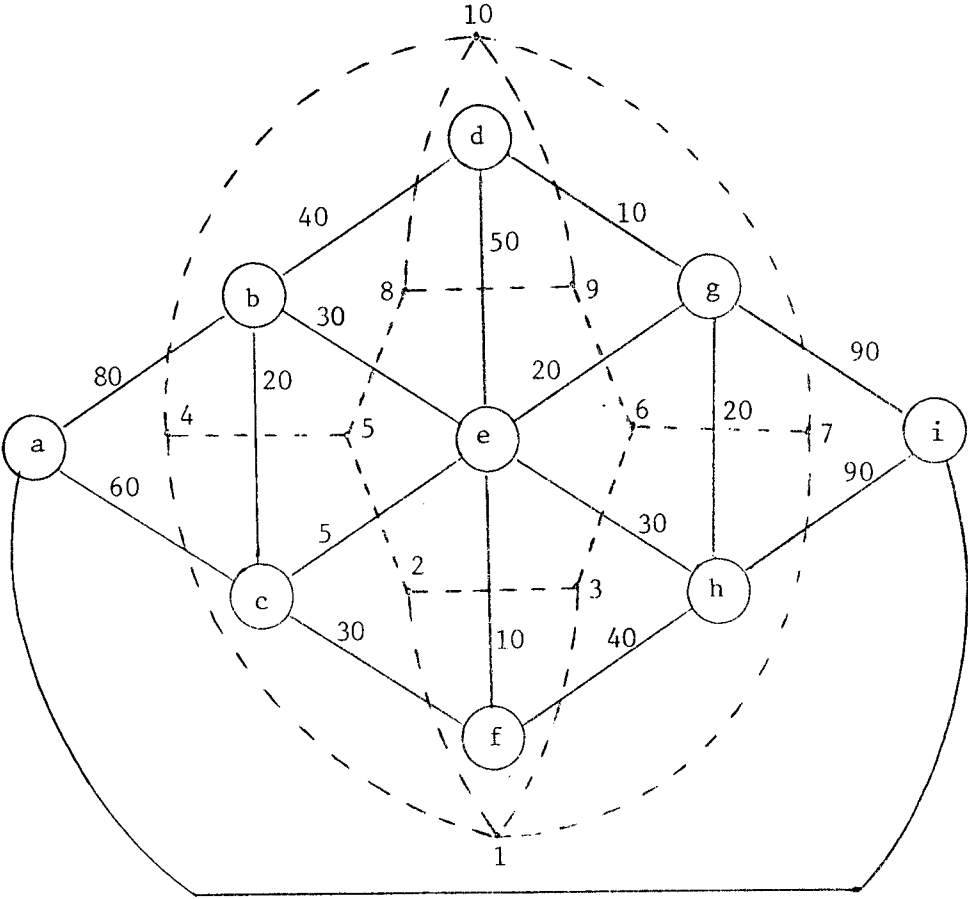


FIGURA 2 - Par dual de redes do exemplo 1.

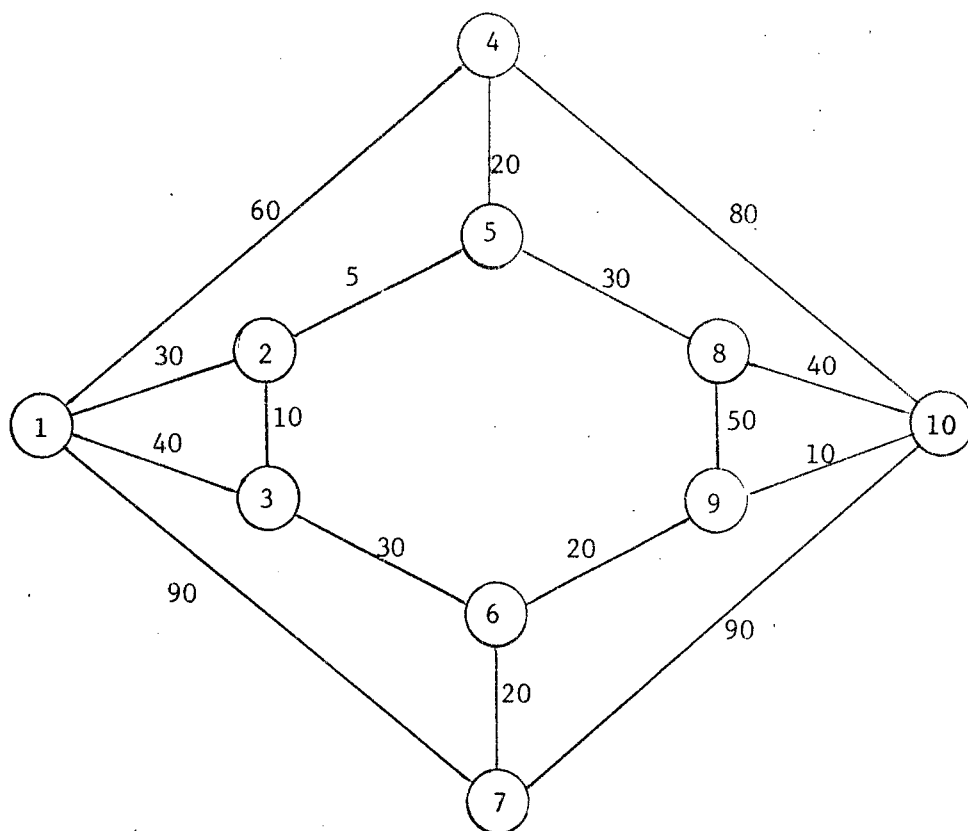


FIGURA 3 - Rede dual apresentada na Figura 2

Aplicando o algoritmo de Wollmer para a rede dual da Figura 3, com $i=0,1,2$, obtêm-se:

M É O NUMERO DE NCS REDE = 10
 N É O NUMERO DE ARCCS A SEREM REDUZIDOS = 2
 NV É O NUMERO DE ARCCS DA REDE= 16

D(2, 0) = 30.00
 D(3, 0) = 40.00
 D(4, 0) = 60.00
 D(7, 0) = 90.00
 D(5, 0) = 35.00
 D(6, 0) = 70.00
 D(4, 0) = 55.00
 D(10, 0) = 135.00
 D(8, 0) = 65.00
 D(9, 0) = 90.00
 D(10, 0) = 105.00
 D(10, 0) = 100.00

O VALOR DESEJADO É D(10,0) = 100.00

$D(2, 1) = 30.00$
 $D(2, 1) = 0.0$
 $D(3, 1) = 40.00$
 $D(3, 1) = 0.0$
 $D(4, 1) = 60.00$
 $D(4, 1) = 0.0$
 $D(7, 1) = 90.00$
 $D(7, 1) = 0.0$
 $D(5, 1) = 5.00$
 $D(6, 1) = 30.00$
 $D(10, 1) = 80.00$
 $D(10, 1) = 55.00$
 $D(8, 1) = 35.00$
 $D(6, 1) = 20.00$
 $D(9, 1) = 40.00$
 $D(10, 1) = 50.00$

O VALOR DESEJADO É $D(10, 1) = 50.00$

$D(2, 2) = 30.00$
 $D(2, 2) = 0.0$
 $D(3, 2) = 40.00$
 $D(3, 2) = 0.0$
 $D(4, 2) = 60.00$
 $D(4, 2) = 0.0$
 $D(7, 2) = 90.00$
 $D(7, 2) = 0.0$
 $D(5, 2) = 5.00$
 $D(5, 2) = 0.0$
 $D(6, 2) = 30.00$
 $D(6, 2) = 0.0$
 $D(10, 2) = 80.00$
 $D(10, 2) = 0.0$
 $D(8, 2) = 30.00$
 $D(8, 2) = 5.00$
 $D(9, 2) = 20.00$
 $D(9, 2) = 10.00$

O VALOR DESEJADO É $D(10, 2) = 0.0$

COM 2 ARCOS FALTANDO O COMPRIMENTO É 0.0

Exemplo 2: Na Figura 4 temos a rede primal (contínua) e sua dual (tracejada). Para a Figura 5 transporta-se a rede dual.

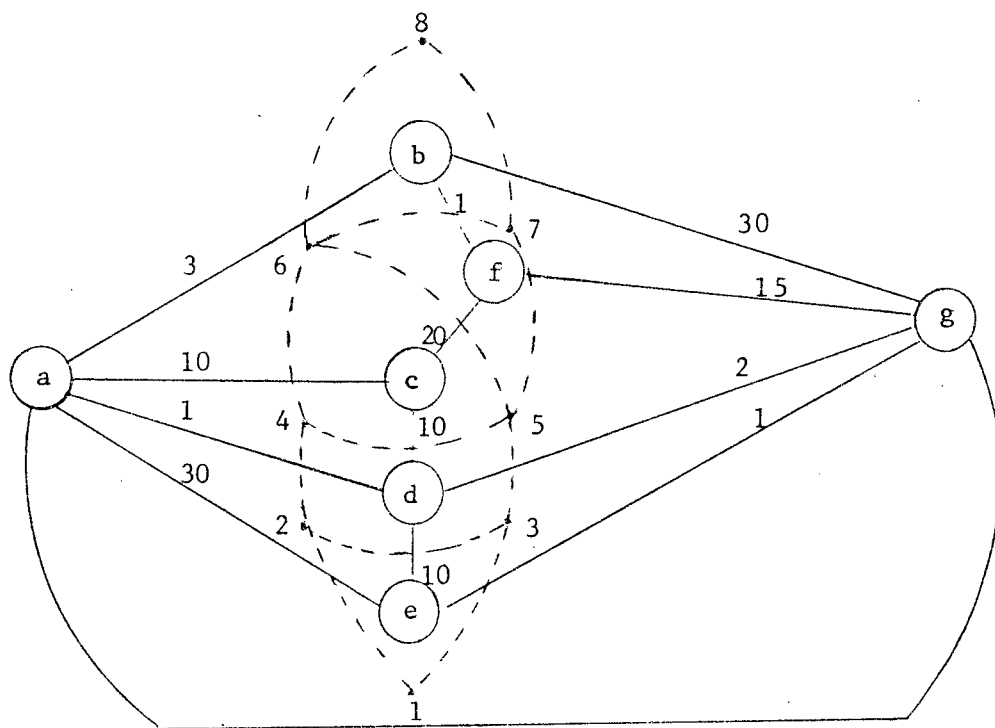


FIGURA 4 - Par dual de redes do exemplo 2.

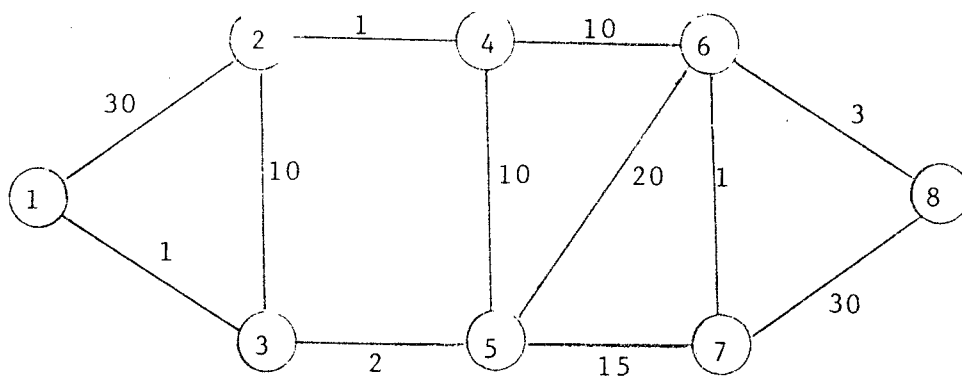


FIGURA 5 - Rede dual apresentada na Figura 4.

Faz-se a aplicação do algoritmo de Wollmer para

$i \in 0,1:$

M É O NÚMERO DE VUS PODE = 2
 N É O NÚMERO DE ARCOS A SEREM REPRODUZIDOS = 1
 NV É O NÚMERO DE ARCOS DA REDE = 12

D(2, 0) = 30.00
 D(3, 0) = 1.00
 D(2, 0) = 11.00
 D(4, 0) = 12.00
 D(5, 0) = 3.00
 D(6, 0) = 22.00
 D(7, 0) = 18.00
 D(6, 0) = 19.00
 D(8, 0) = 22.00

O VALOR DESEJADO É D(8,0) = 22.00

D(2, 1) = 30.00
 D(2, 1) = 0.0
 D(3, 1) = 1.00
 D(3, 1) = 0.0
 D(4, 1) = 1.00
 D(5, 1) = 2.00
 D(5, 1) = 1.00
 D(6, 1) = 11.00
 D(6, 1) = 3.00
 D(7, 1) = 16.00
 D(7, 1) = 3.00
 D(8, 1) = 6.00

O VALOR DESEJADO É D(8,1) = 6.00

CCM 1 ARCOS FALHANDO O COMPRIMENTO É 6.000

3.2. Modificações

Fazendo considerações sobre a entrada de dados e observando que em algumas redes o valor do caminho mais curto da fonte até algum nó intermediário não era obtido corretamente, algumas modificações foram introduzidas no algoritmo de Wollmer.

Primeira Modificação

Refere-se à inicialização das variáveis. Quando $k=0$, far-se-á $D_{s,0}=0$ e $D_{a,0}=\infty$ ($\forall a \neq s$). Depois de obtidos todos os

$D_{a,k} = L_{a,k}$ far-se-á $k=k+1$ e $D_{a,k} = D_{a,k-1}$, isto é, a cada acréscimo de k as variáveis D serão inicializadas com os valores anteriores. Isto pode ser justificado, considerando que um caminho viável de s a qualquer nó a com k arcos falhando é $D_{a,k-1}$, pois $D_{a,k} \leq D_{a,k-1}$. O algoritmo ficará:

1. Para $i=0$ faça $D_{s,i}=0$ e $D_{a,i}=\infty$ ($\forall a \neq s$)

$k=0$

Executa-se o passo 2 e

3. Então:

a) Se $k < n$ faça $k=k+1$ e $D_{a,k} = D_{a,k-1}$ ($\forall a$)
e vá ao passo 2.

b) Se $k=n$ termine, com $D_{t,n}$ sendo o valor desejado.

Segunda Modificação:

Observando que, ao satisfazer a inequação $D_{a,k} > D_{b,k} + \ell(a,b)$, então $D_{a,k} = D_{b,k} + \ell(a,b)$, quando $b=t$, teremos um caminho que sai da fonte passa pelo sumidouro e chega até o nó a . Foi incluída, então, a restrição de que $b \neq m$ nesta inequação. Assim, no algoritmo:

2. a) Se $D_{a,k} > D_{b,k} + \ell(a,b)$ e $b < t$ faça $D_{a,k} = D_{b,k} + \ell(a,b)$

Considerando que os nós da rede são numerados de tal forma que s é o primeiro e t é o último; a entrada de dados é feita; $a, b, \ell(a,b)$, isto é, cada arco (a,b) e seu valor $\ell(a,b)$, sendo que $\forall a,b, a < b$.

Terceira Modificação

Verifica-se que, em geral, $D_{b,k} > D_{a,k} + \ell(a,b)$ e quan

do $k > 0$, $D_{b,k} > D_{a,k-1}$. Ocorre também que se $D_{b,k} > D_{a,k} + \ell(a,b)$ então $D_{b,k} = D_{a,k} + \ell(a,b)$, portanto não será possível que $D_{a,k} > D_{b,k} + \ell(a,b)$ para os mesmos valores de a e b ; e se $D_{b,k} > D_{a,k-1}$ então $D_{b,k} = D_{a,k-1}$ e não será possível $D_{a,k} > D_{b,k-1}$. Verifica-se ainda que só uma substituição de $D_{a,k} = D_{b,k} + \ell(a,b)$ fará necessária uma nova iteração de pesquisa do caminho mais curto, pois a partir do nó a outros poderão ser alcançados com distâncias menores. Quando a substituição é feita para o nó b , os nós a serem alcançados a partir dele ainda serão pesquisados na mesma iteração.

No algoritmo:

2. Para cada arco (a,b) , verifique:

- a) Se $D_{b,k} > D_{a,k-1}$ faça $D_{b,k} = D_{a,k-1}$ e vá a c)
- b) Se $D_{a,k} > D_{b,k-1}$ faça $D_{a,k} = D_{b,k-1}$
- c) Se $D_{b,k} > D_{a,k} + \ell(a,b)$ faça $D_{b,k} = D_{a,k} + \ell(a,b)$ e vá ao próximo arco (a,b)
- d) Se $D_{a,k} > D_{b,k} + \ell(a,b)$ faça $D_{a,k} = D_{b,k} + \ell(a,b)$

Até que nenhuma mudança em d) seja verificada.

Quarta Modificação:

O problema de determinar o valor do caminho mais curto quando n arcos são removidos, se $n=0$ torna-se o problema clássico de obtenção do valor do caminho mais curto e é resolvido pelo algoritmo sem os itens: 2c), 2d), 3a). Para que não seja necessário questionar sempre o valor de n , o passo 2 do algoritmo será desmembrado em duas partes:

Seja $k=0$

2. Para cada arco (a,b), verifique:

- a) Se $D_{b,k} > D_{a,k} + \ell(a,b)$ faça $D_{b,k} = D_{a,k} + \ell(a,b)$ e vá ao próximo arco (a,b)
- b) Se $D_{a,k} > D_{b,k} + \ell(a,b)$ faça $D_{a,k} = D_{b,k} + \ell(a,b)$ e vá ao próximo arco (a,b)

Até que nenhuma mudança em b) seja verificada.

Se $k=n$, termine.

Senão, faça $k=k+1$

3. Para cada arco (a,b), verifique:

- a) Se $D_{b,k} > D_{a,k-1}$ faça $D_{b,k} = D_{a,k-1}$ e vá para c)
- b) Se $D_{a,k} > D_{b,k-1}$ faça $D_{a,k} = D_{b,k-1}$ e vá para c)
- c) Se $D_{b,k} > D_{a,k} + \ell(a,b)$ faça $D_{b,k} = D_{a,k} + \ell(a,b)$ e vá ao próximo arco (a,b)
- d) Se $D_{a,k} > D_{b,k} + \ell(a,b)$ faça $D_{a,k} = D_{b,k} + \ell(a,b)$ e vá ao próximo arco (a,b).

Até que nenhuma mudança em d) seja verificada.

3.2.1. Exemplos do algoritmo com modificações

Exemplo 3: Para o mesmo problema do exemplo 1, cuja rede é apresentada nas Figuras 2 e 3, com as modificações mencionadas, obteve-se a saída:

M É O NUMERO DE VCS REDES = 10
 N É O NUMERO DE ARCCS A SEREM REDUZIDOS = 2
 NV É O NUMERO DE ARCCS DA REDE = 16

D(2, 0) = 30.00
 D(3, 0) = 40.00
 D(4, 0) = 60.00
 D(7, 0) = 90.00

$D(5, 0) = 35.00$
 $D(6, 0) = 70.00$
 $D(4, 0) = 55.00$
 $D(10, 0) = 135.00$
 $D(8, 0) = 65.00$
 $D(9, 0) = 90.00$
 $D(10, 0) = 105.00$
 $D(10, 0) = 100.00$

O VALOR DESAJADO É $D(10, 0) = 100.00$

$D(2, 1) = 0.0$
 $D(3, 1) = 0.0$
 $D(4, 1) = 0.0$
 $D(7, 1) = 0.0$
 $D(5, 1) = 30.00$
 $D(5, 1) = 5.00$
 $D(6, 1) = 40.00$
 $D(6, 1) = 30.00$
 $D(10, 1) = 55.00$
 $D(8, 1) = 35.00$
 $D(6, 1) = 20.00$
 $D(9, 1) = 70.00$
 $D(9, 1) = 40.00$
 $D(10, 1) = 50.00$

O VALOR DESAJADO É $D(10, 1) = 50.00$

$D(5, 2) = 0.0$
 $D(6, 2) = 0.0$
 $D(10, 2) = 0.0$
 $D(3, 2) = 5.00$
 $D(9, 2) = 20.00$

O VALOR DESAJADO É $D(10, 2) = 0.0$

COM 2 ARCOS FALHADOS E COMPRIMENTO É 0.0

Exemplo 4: Aqui a rede é igual à do exemplo 2, apresentada nas Figuras 4 e 5. A resposta foi:

M É O NUMERO DE NOS REDE = 8
 N É O NUMERO DE ARCOS A SEREM REDUZIDOS = 1
 NV É O NUMERO DE ARCOS DA REDE = 12

$D(2, 0) = 30.00$
 $D(3, 0) = 1.00$
 $D(2, 0) = 11.00$
 $D(4, 0) = 12.00$
 $D(5, 0) = 3.00$
 $D(6, 0) = 22.00$
 $D(7, 0) = 18.00$

D(6, 0) = 19.00
 D(3, 0) = 22.00

O VALOR DESEJADO É D(8,0) = 22.00

D(2, 1) = 0.0
 D(3, 1) = 0.0
 D(4, 1) = 11.00
 D(4, 1) = 1.00
 D(5, 1) = 1.00
 D(6, 1) = 12.00
 D(6, 1) = 11.00
 D(6, 1) = 3.00
 D(7, 1) = 3.00
 D(8, 1) = 19.00
 D(8, 1) = 6.00

O VALOR DESEJADO É D(8,1) = 6.00

COM 1 ARCOS FALHANDO O COMPRIMENTO É 6.00

Pode-se observar nestes exemplos que as modificações introduzidas reduziram o número de cálculos efetuados, já que a parecem impressas todas as mudanças de valor de $D_{a,k}$.

3.3. Comparação do Algoritmo de Wollmer com o Algoritmo Modificado

Para efetuar uma comparação dos tempos de execução dos dois algoritmos: o original e o modificado, era necessário que os dois fornecessem as mesmas respostas. Por isso considerou-se a segunda modificação também no algoritmo original. Os algoritmos foram programados em FORTRAN. Foram executados testes em redes de diversos tamanhos, com diferentes tipos de conexidade, para 2 e 4 falhas, no computador IBM 4341 da Universidade Federal de Santa Catarina. Os valores dos arcos foram obtidos através de uma geração de números aleatórios uniformemente distribuídos em

torno de uma média especificada. Os resultados obtidos estão mostrados nas tabelas abaixo, onde M é o número de nós e NA o número de arcos. As redes a) e b) são simplesmente conexas, com os valores dos arcos sendo variáveis aleatórias uniformemente distribuídas com média 20, testadas para 2 falhas. As redes c) e d) são fortemente conexas, com os valores dos arcos gerados com média 10, sendo testadas para 2 e 4 falhas.

Na tabela 1 são consideradas redes cujas duais possuem 10 nós. São apresentadas na Figura 6.

$M = 10$						
	a) NA=12	b) NA=13	c) NA=19		d) NA=21	
	n=2	n=2	n=2	n=4	n=2	n=4
Original $\times 10^{-2}s$	21,83	22,3	24,0	25,5	24,33	27,0
Modificado $\times 10^{-2}s$	21,67	21,67	23,67	24,67	24,0	26,5

TABELA 1 - Comparação dos tempos de execução de redes com 10 nós.

Na tabela 2 as redes testadas possuem 30 nós. Estas estão apresentadas na Figura 7.

$M = 30$						
	a) NA=40	b) NA=43	c) NA=66		d) NA=77	
	n=2	n=2	n=2	n=4	n=2	n=4
Original $\times 10^{-2}s$	32,33	35,17	41,17	46,67	47,33	51,67
Modificado $\times 10^{-2}s$	30,67	33,83	40,83	44,17	46,5	50,67

TABELA 2 - Comparação aos tempos de execução de redes com 30 nós.

Os testes para redes com 50 nós são apresentados na tabela 3. Suas representações estão na Figura 8.

	M = 50					
	a) NA=71	b) NA=72	c) NA=129		d) NA=131	
	n=2	n=2	n=2	n=4	n=2	n=4
Original x 10^{-2} s	46,83	57,5	60,67	70,5	68,67	78,83
Modificado x 10^{-2} s	46,17	56,67	58,5	64,67	67,33	74,5

TABELA 3 - Comparação dos tempos de execução de redes com 50 nós.

Para redes com 100 nós os resultados estão na tabela 4. Suas configurações aparecem na Figura 9.

	M = 100					
	a) NA=147	b) NA=157	c) NA=271		d) NA=271	
	n=2	n=2	n=2	n=4	n=2	n=4
Original x 10^{-2} s	107,33	120,67	125,0	152,33	183,67	217,0
Modificado x 10^{-2} s	108,5	120,67	123,0	148,5	170,17	203,67

TABELA 4 - Comparação dos tempos de execução de redes com 100 nós

Os tempos de execução para redes de 500 nós estão apresentados na tabela 5. As redes testadas aparecem na Figura 10.

	M = 500					
	a) NA=598	b) NA=715	c) NA=1426		d) NA=1437	
	n=2	n=2	n=2	n=4	n=2	n=4
Original x 10^{-2} s	5425	4048	4146,2	6097,2	3402,3	5202,6
Modificado x 10^{-2} s	5070,5	3932,7	3561	5577,7	3395,8	5179,8

TABELA 5 - Comparação dos tempos de execução de redes com 500 nós.

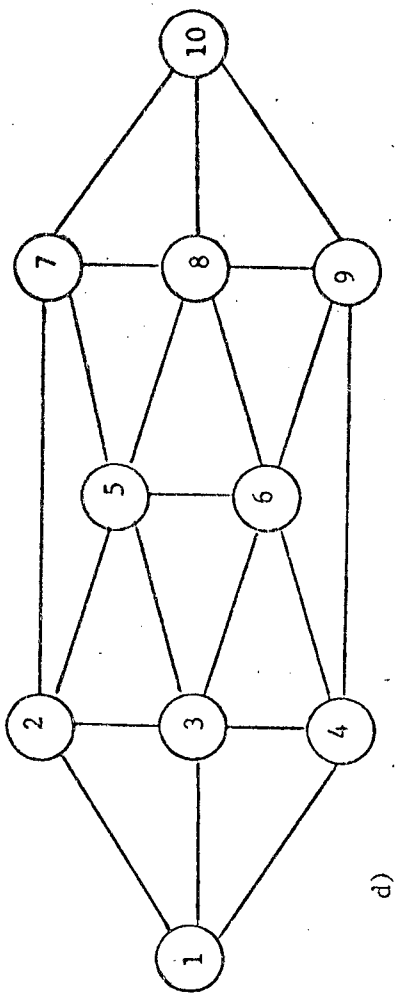
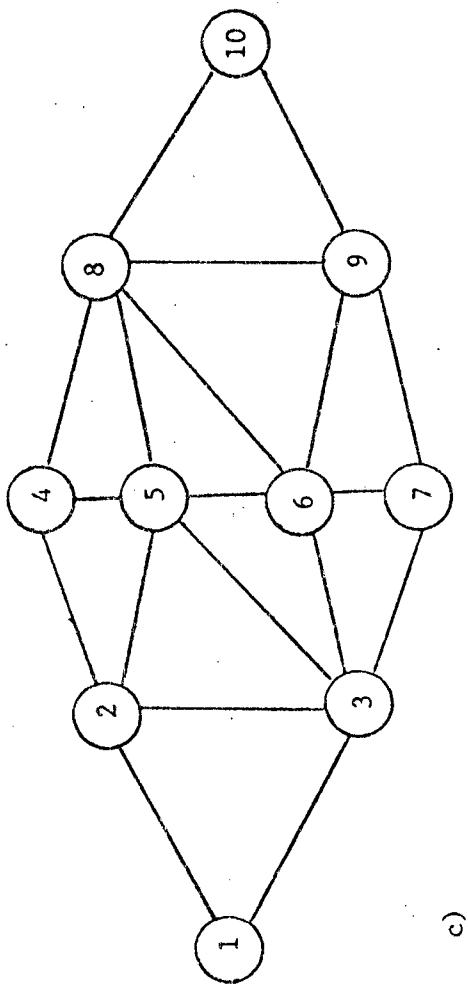
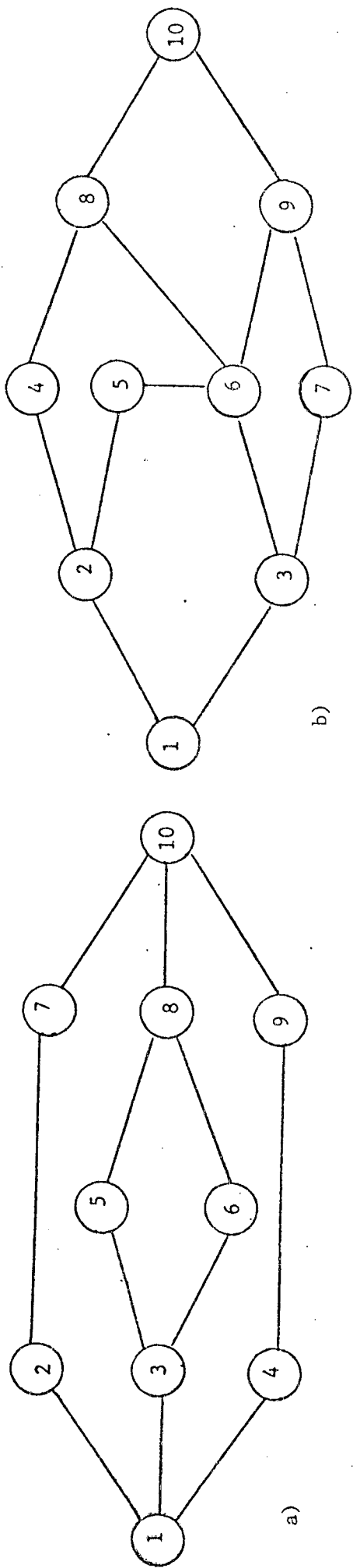


FIGURA 6 - Redes com 10 nós.

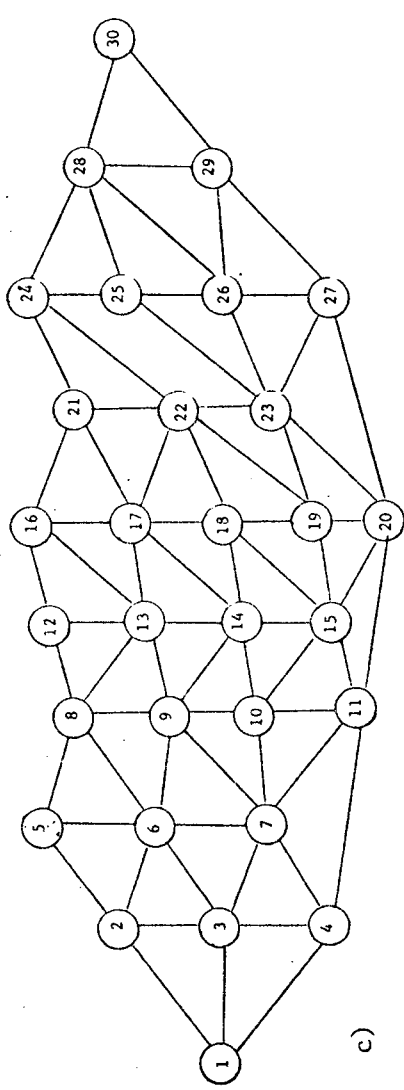
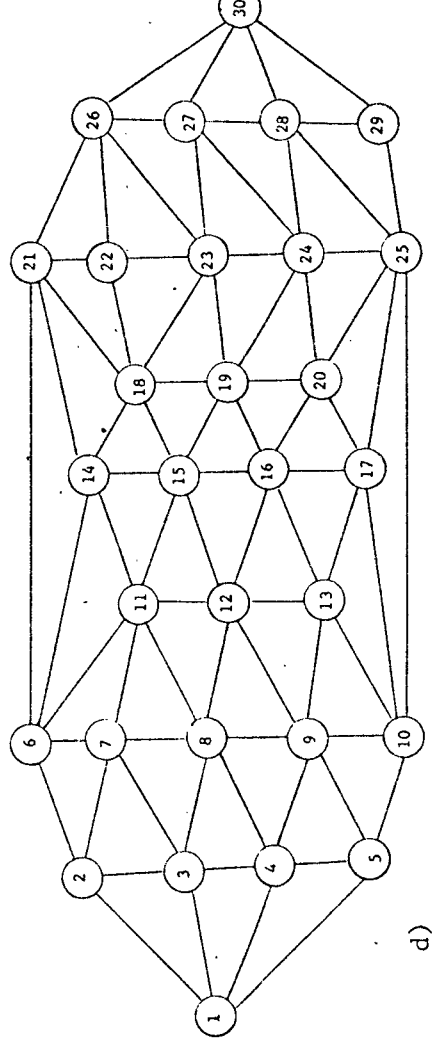
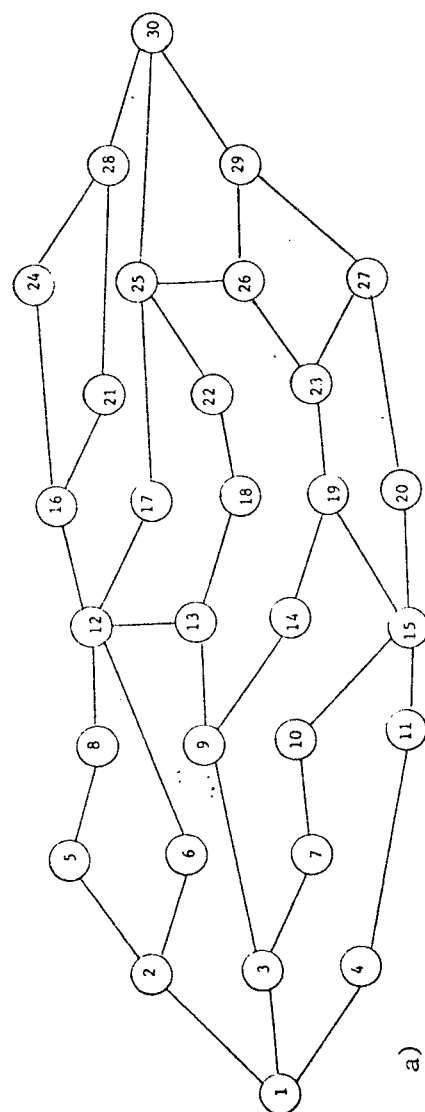
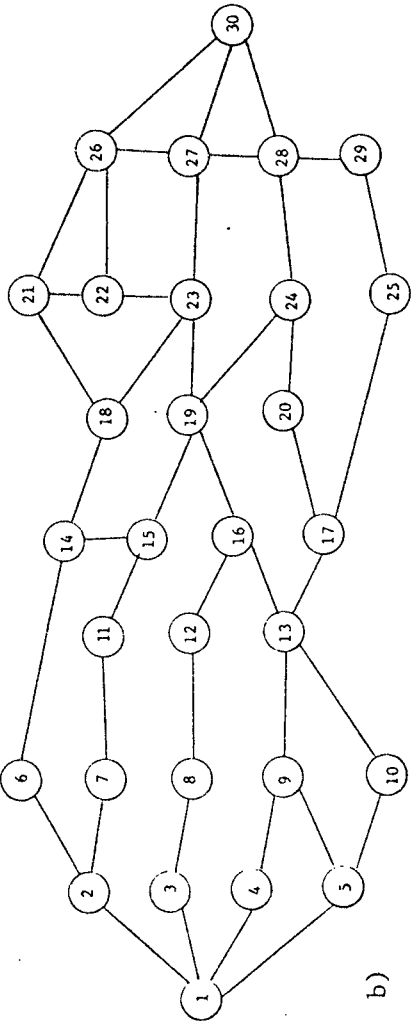
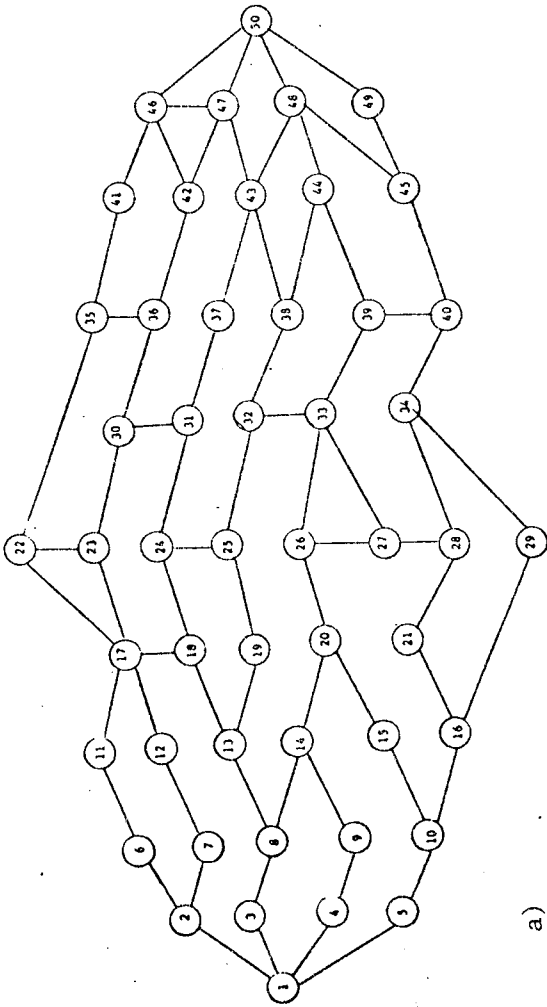
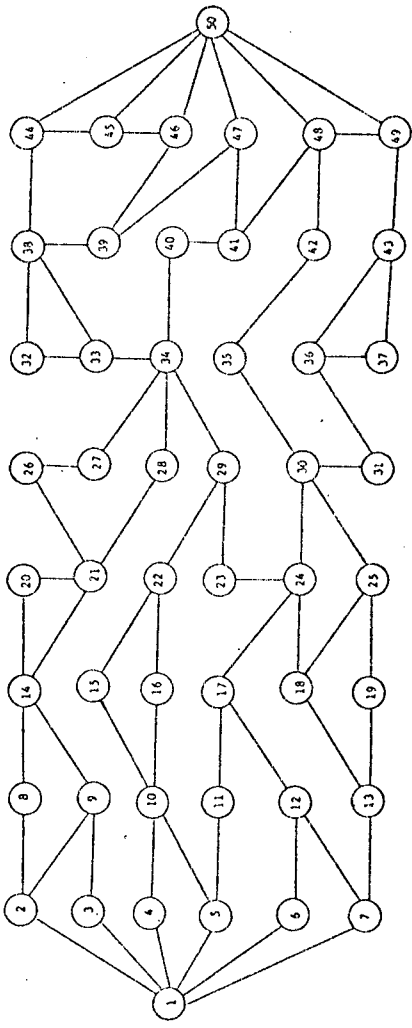


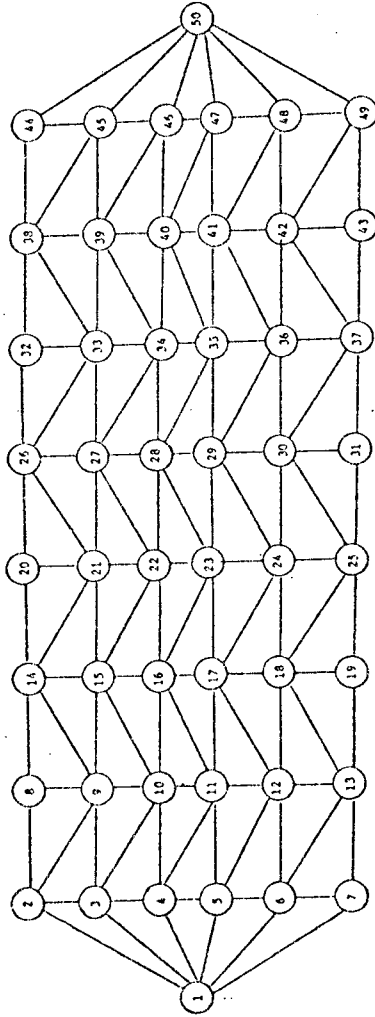
FIGURA 7 - Redes com 30 nós.



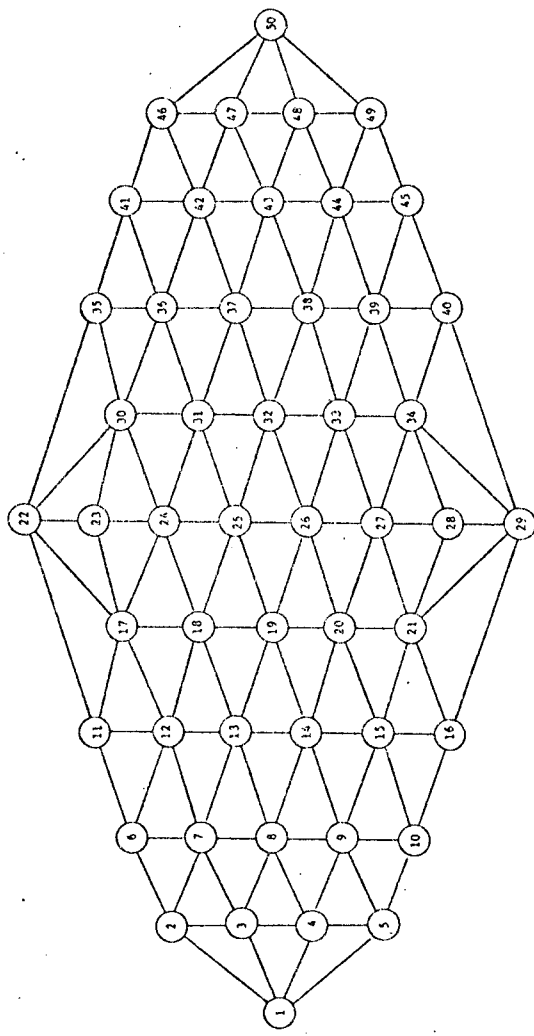
a)



b)

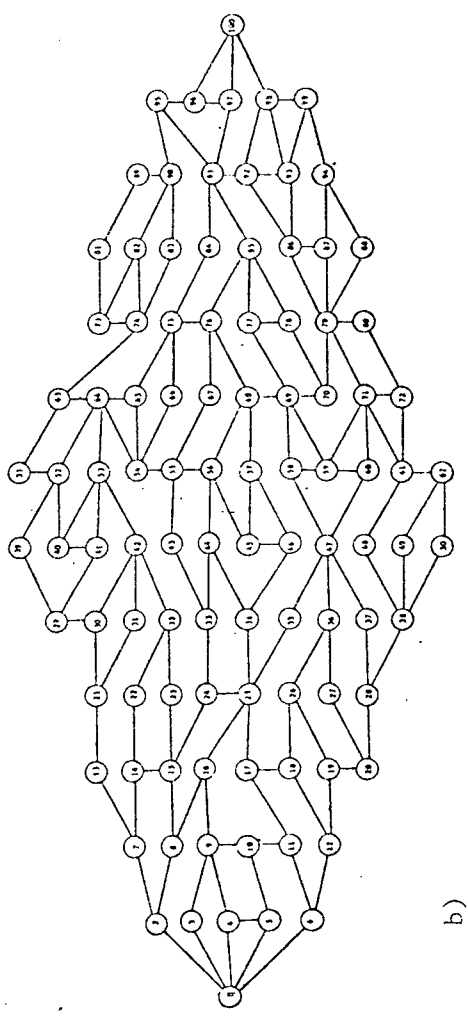


c)

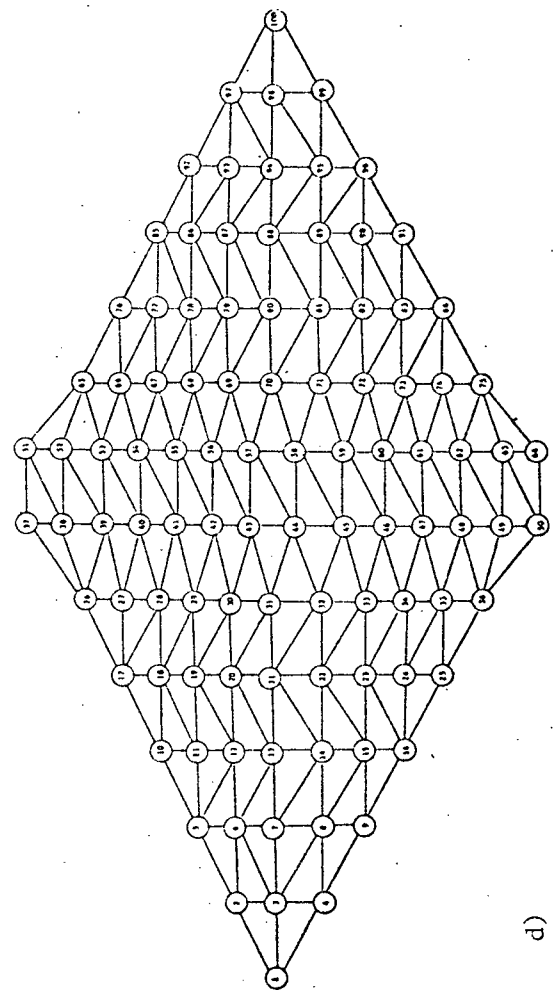


d)

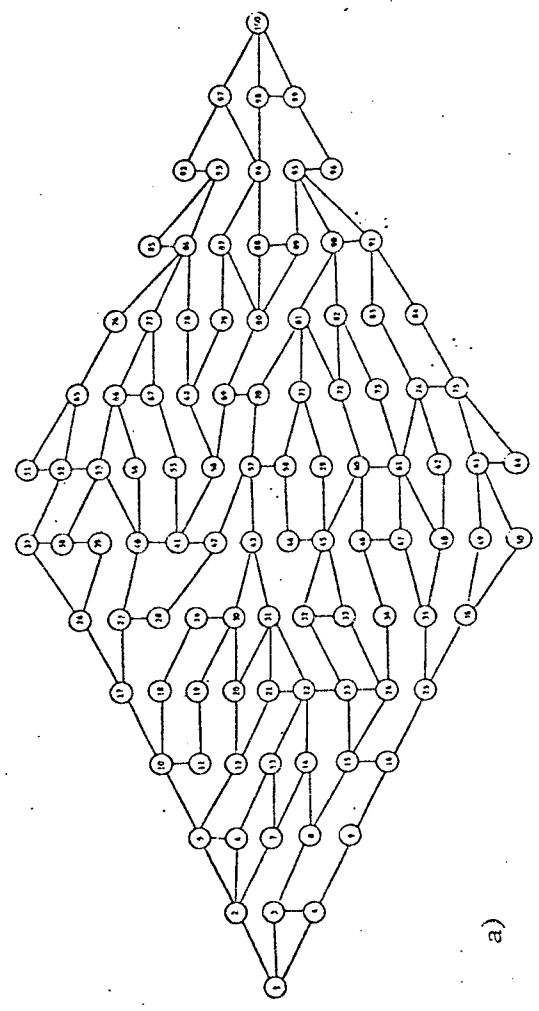
FIGURA 8 - Redes com 50 nós.



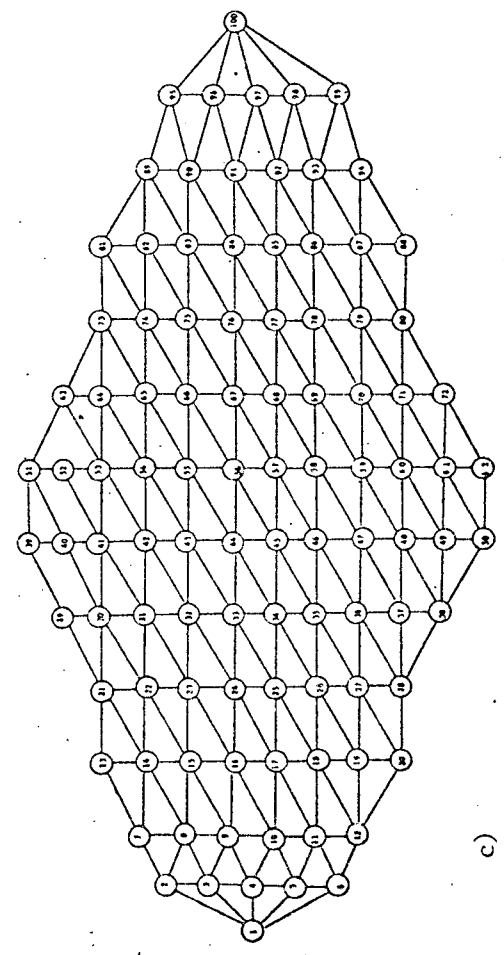
b)



d)



a)



c)

FIGURA 9 - Redes com 100 nós.

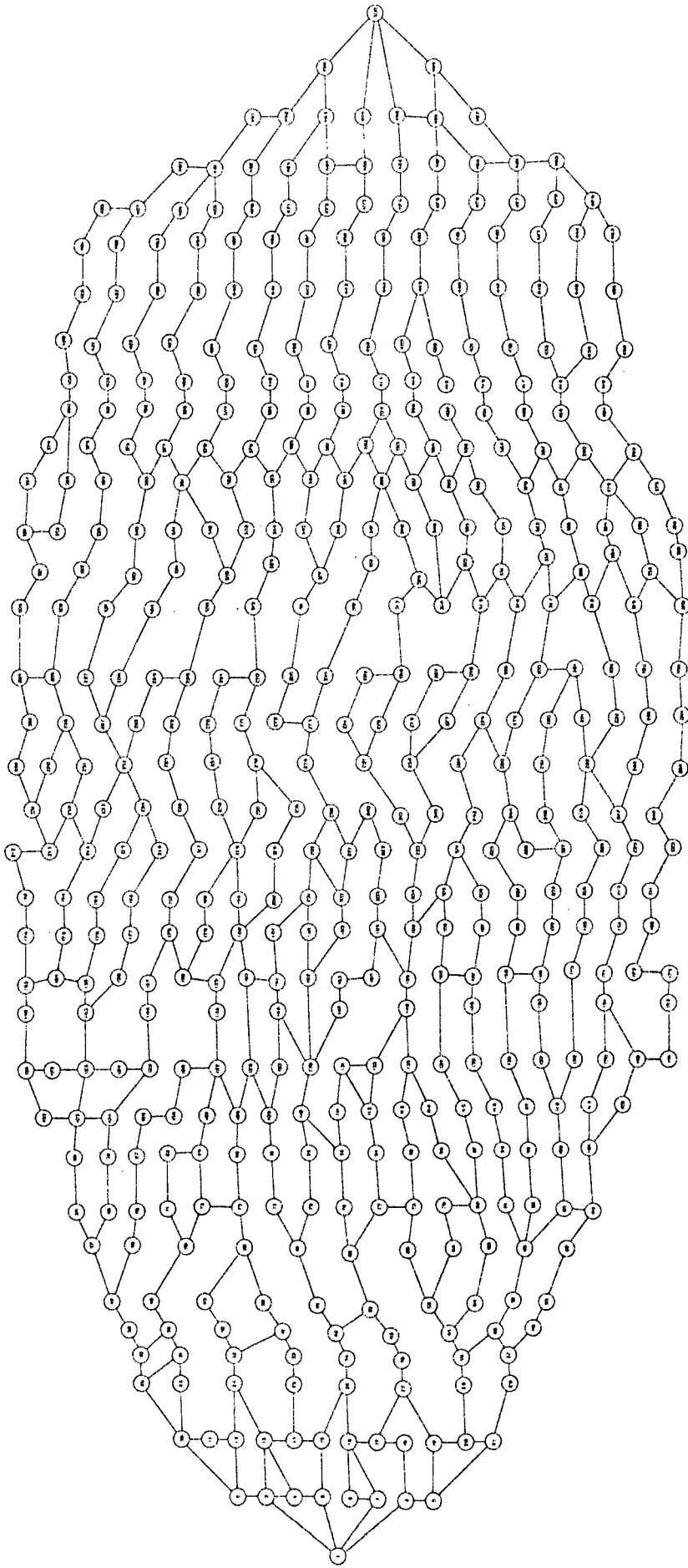


FIGURA 10a - Rede com 500 nós.

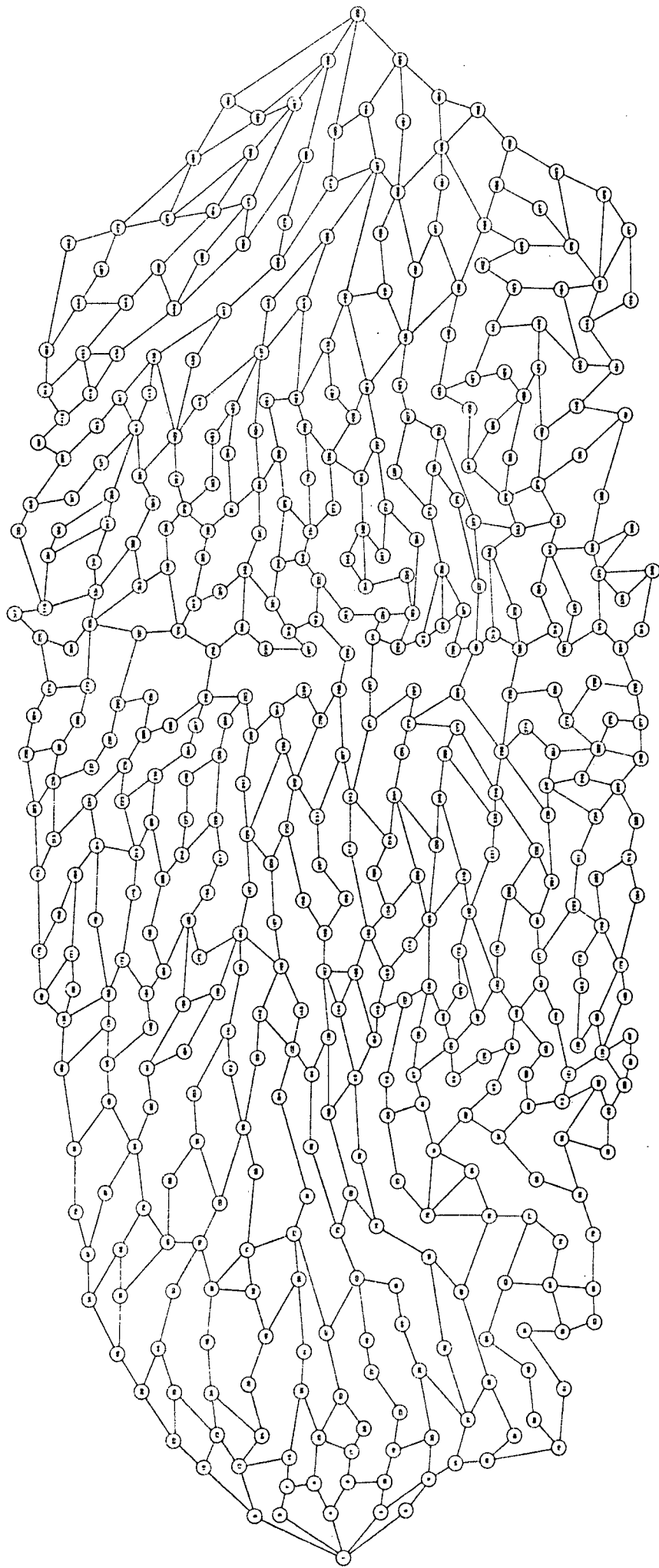


FIGURA 10b - Rede com 500 nós.

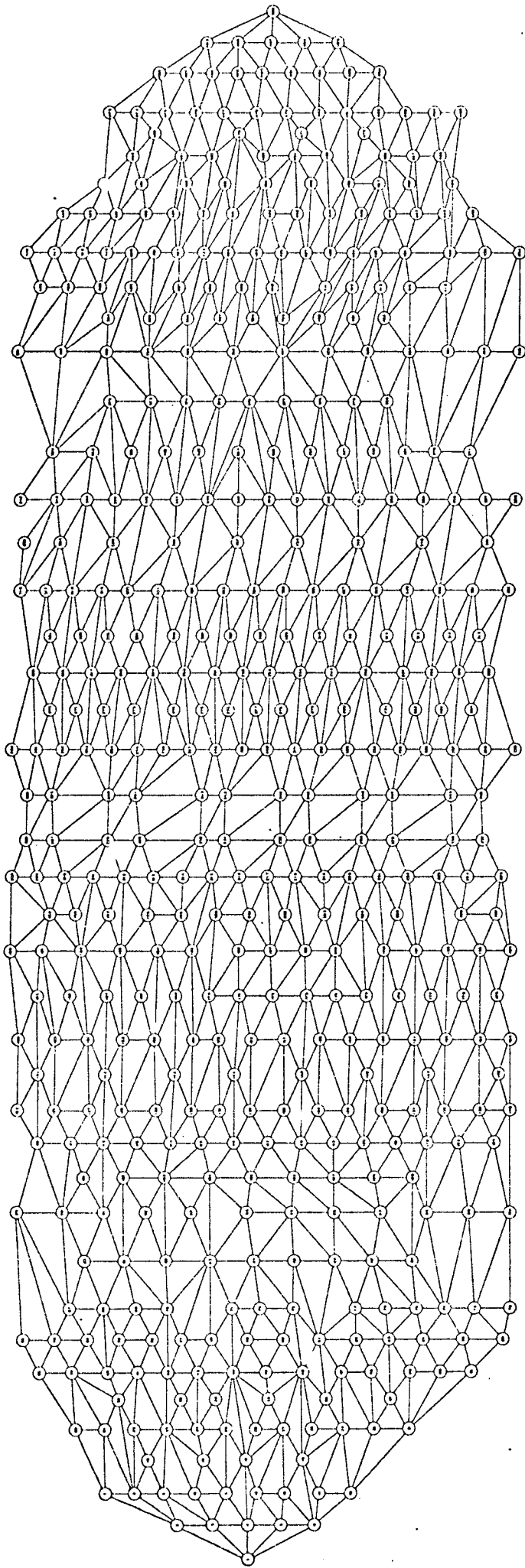


FIGURA 10c - Rede com 500 nós.

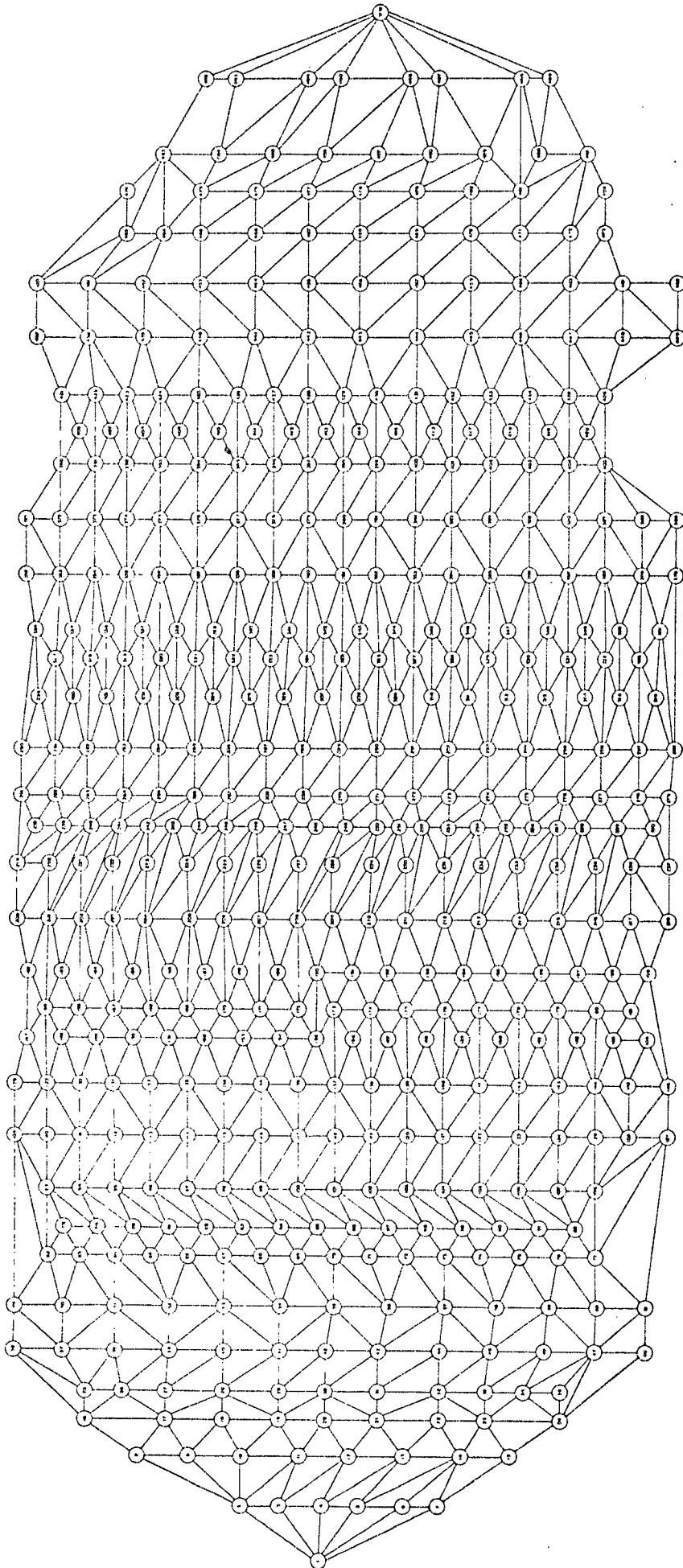


FIGURA 10d - Rede com 500 nós.

Temos que, em geral, o algoritmo modificado executa o processo mais rápido que o original. Pode-se verificar que isto se acentua à medida que o número de falhas aumenta.

3.4. Segunda Parte do Algoritmo de Wollmer: Determinação do Caminho mais Curto

Tendo obtido o valor do caminho mais curto da fonte até o sumidouro, Wollmer apresenta um algoritmo para a obtenção do caminho mais curto e para a indicação dos arcos que falharam.

O procedimento inicia no sumidouro e pesquisa os nós na direção da fonte, a procura dos nós que compõem o caminho. No princípio $i=n$, isto é, o número máximo de falhas e depois vai diminuindo.

O algoritmo é o seguinte:

1. Seja $k=1$, $a(1)=t$ e $i(1)=n$.
2. Encontre um nó $a(k+1)$, tal que:

$$D_{a(k+1), i(k+1)} = D_{a(k), i(k)} - \ell[a(k), a(k+1)] \quad \text{e} \\ i(k+1) = i(k)$$

ou $D_{a(k+1), i(k+1)} = D_{a(k), i(k)}$ e $i(k+1) = i(k) - 1$.

3. Se $a(k+1) \neq s$, faça $k=k+1$ e vá a 2.

Senão, termine.

O caminho desejado é $s=a(k+1), \dots, a(1)=t$.

Os arcos que falharam foram aqueles $[a(k), a(k+1)]$ tais que $i(k) \neq i(k+1)$.

Os arcos da rede primal que devem ser removidos são aqueles que interceptam estes arcos da rede dual.

Sendo: s - fonte

t - sumidouro

k - contador

a - nós pertencentes ao caminho

i - quantidade de arcos que podem sofrer redução.

Exemplo 5: Considerando a rede dual como a da Figura 6a), com os valores correspondentes sendo a,b, $\ell(a,b)$, então:

1	2	5.
1	3	7.
1	4	27.
2	7	29.
3	5	12.
3	6	39.
4	9	19.
5	8	35.
6	8	6.
7	10	28.
8	10	3.
9	10	33.

M É O NUMERO DE NOS REDE = 10

N É O NUMERO DE ARCOS A SEREM REDUZIDOS = 2

NV É O NUMERO DE ARCOS DA REDE= 12

D(2, C)=	6.00000
D(3, C)=	7.00000
D(4, C)=	27.00000
D(7, C)=	35.00000
D(5, C)=	19.00000
D(6, C)=	46.00000
D(9, C)=	46.00000
D(8, C)=	54.00000
D(8, C)=	52.00000
D(10, C)=	63.00000
D(10, C)=	55.00000

A ROTA DESEJADA E D(10,0)= 54.00

D(2, 1)= 6.00000
 D(2, 1)= 0.0
 D(3, 1)= 7.00000
 D(3, 1)= 0.0
 D(4, 1)= 27.00000
 D(4, 1)= 0.0
 D(7, 1)= 27.00000
 D(7, 1)= 5.00000
 D(5, 1)= 12.00000
 D(5, 1)= 7.00000
 D(6, 1)= 39.00000
 D(6, 1)= 7.00000
 D(9, 1)= 19.00000
 D(8, 1)= 42.00000
 D(8, 1)= 19.00000
 D(8, 1)= 13.00000
 D(10, 1)= 34.00000
 D(10, 1)= 16.00000

A ROTA DESEJADA E D(10,1)= 16.00

D(2, 2)= 6.00000
 D(2, 2)= 0.0
 D(3, 2)= 7.00000
 D(3, 2)= 0.0
 D(4, 2)= 27.00000
 D(4, 2)= 0.0
 D(7, 2)= 29.00000
 D(7, 2)= 0.0
 D(5, 2)= 12.00000
 D(5, 2)= 0.0
 D(6, 2)= 39.00000
 D(6, 2)= 0.0
 D(9, 2)= 19.00000
 D(9, 2)= 0.0
 D(8, 2)= 35.00000
 D(8, 2)= 7.00000
 D(8, 2)= 6.00000
 D(10, 2)= 28.00000
 D(10, 2)= 6.00000

A ROTA DESEJADA E D(10,2)= 6.00

COM 2 ARCOS FALHANDO O COMPLEMENTO E 6.00

O CAMINHO MINIMO PARA 2 ARCOS E

10 7 2 1

AS REDUÇÕES OCORRERAM NOS ARCOS

10	7
7	2
2	1

O CAMINHO MINIMO PARA 1 ARCOS É

10 6 3 1

AS REDUÇÕES OCORRERAM NOS ARCOS

6	3
3	1

O CAMINHO MINIMO PARA 0 ARCOS É

10 6 3 1

Este procedimento é verdadeiro em algumas situações, mas dependendo da configuração da rede e dos valores dos arcos, não fornece os resultados esperados. Pode-se observar no exemplo 5, que o caminho mais curto e seu valor foram obtidos corretamente, mas na resolução do problema com possibilidade de um arco falhar, foram apontados dois arcos. O arco que une o nó 1 ao nó 3 sofreu redução no caminho mais curto da fonte até o nó 3; o arco que une o nó 3 ao nó 6 sofreu redução no caminho mais curto da fonte até o nó 6; assim o arco que falhou é o de maior comprimento. O mesmo ocorre para o processo com duas falhas.

A seguir, é dado um exemplo que mostra que este processo também não é muito seguro para a determinação do caminho mais curto em uma rede.

Exemplo 6: Seja a rede dual apresentada na Figura 6c), pelo algoritmo de Wollmer, com os valores dos arcos sendo os seguintes:

a	b	$\ell(a,b)$
1	2	1.
1	3	19.
2	3	19.
2	4	18.
2	5	15.
3	5	5.
3	6	6.
3	7	8.
4	5	18.
4	8	6.
5	6	19.
5	8	8.
6	7	12.
6	8	3.
6	9	4.
7	9	13.
8	9	14.
8	10	6.
9	10	19.

M É O NÚMERO DE NÓS REDE = 10

N É O NÚMERO DE ARCOS A SEREM REDUZIDOS = 2

NV É O NÚMERO DE ARCOS DA REDE = 19

D(2, C) =	1.00000
D(3, C) =	19.00000
D(4, C) =	19.00000
D(5, C) =	16.00000
D(6, C) =	25.00000
D(7, C) =	27.00000
D(8, C) =	25.00000
D(8, C) =	24.00000
D(9, C) =	29.00000
D(10, C) =	30.00000

A PONTA DESEJADA É D(10,0) = 30.00

D(2, 1) =	1.00000
D(2, 1) =	0.0
D(3, 1) =	19.00000
D(3, 1) =	0.0
D(4, 1) =	18.00000
D(4, 1) =	1.00000
D(5, 1) =	15.00000
D(5, 1) =	1.00000

$D(6, 1) = 6.00000$
 $D(7, 1) = 8.00000$
 $D(8, 1) = 7.00000$
 $D(9, 1) = 10.00000$
 $D(10, 1) = 13.00000$

A ROTA DESEJADA É $D(10, 1) = 13.00$

$D(2, 2) = 1.00000$
 $D(2, 2) = 0.0$
 $D(3, 2) = 19.00000$
 $D(3, 2) = 0.0$
 $D(4, 2) = 12.00000$
 $D(4, 2) = 0.0$
 $D(5, 2) = 15.00000$
 $D(5, 2) = 0.0$
 $D(6, 2) = 6.00000$
 $D(6, 2) = 0.0$
 $D(7, 2) = 8.00000$
 $D(7, 2) = 0.0$
 $D(8, 2) = 6.00000$
 $D(8, 2) = 1.00000$
 $D(9, 2) = 4.00000$
 $D(10, 2) = 7.00000$

A ROTA DESEJADA É $D(10, 2) = 7.00$

COM 2 ARCOS FALHANDO O COMPROMISSO É 7.00

O CAMINHO MINIMO PARA 2 ARCOS É

10 8 5 3 2 1

AS RELOCUES OCORRERAM NOS ARCOS

8	5
5	3
3	2
2	1

O CAMINHO MINIMO PARA 1 ARCO É

10 6 4 2 1

AS RELOCUES OCORRERAM NOS ARCOS

4	2
2	1

O CAMINHO MINIMO PARA 0 ARCOS É

10 8 5 2 1

O caminho mais curto para nenhuma falha está correto, pois:

$$\ell(1,2) = 1$$

$$\ell(2,5) = 15$$

$$\ell(5,8) = 8$$

$$\ell(8,10) = 6, \text{ totalizando } D(10,0)=30$$

O caminho mais curto para uma falha também está correto:

$$\ell(1,2) = 1$$

$$\ell(2,4) = 18$$

$$\ell(4,8) = 6$$

$$\ell(8,10) = 6, \text{ suprimindo } \ell(2,4)=18, \text{ que é o maior,} \\ \text{então } D(10,1)=13.$$

O caminho mais curto para duas falhas:

$$\ell(1,2) = 1$$

$$\ell(2,3) = 19$$

$$\ell(3,5) = 5$$

$$\ell(5,8) = 8$$

$$\ell(8,10) = 6, \text{ suprimindo } \ell(2,3)=19 \text{ e } \ell(5,8)=8, \\ D(10,2)=12, \text{ que não está correto.}$$

Para mostrar como estes valores foram obtidos, pode-se acompanhar a segunda parte do algoritmo de Wollmer.

Tendo $D(10,2)=7$ inicia-se a pesquisa do caminho fazendo:

$$k=1$$

$$a(1)=10$$

$$i(1)=2$$

Encontra-se um $\tilde{n\circ}$ $a(2)=8$, tal que:

$$D_{8,2} = D_{10,2} - \ell(8,10)$$

$$1 = 7 - 6$$

Encontra-se um $\tilde{n\circ}$ $a(3)=5$, tal que:

$$D_{5,1} = D_{8,2}$$

$$1 = 1$$

Encontra-se um $\tilde{n\circ}$ $a(4)=3$, tal que:

$$D_{3,1} = D_{5,2}$$

$$0 = 0$$

Encontra-se um $\tilde{n\circ}$ $a(5)=2$, tal que:

$$D_{2,1} = D_{3,2}$$

$$0 = 0$$

Encontra-se um $\tilde{n\circ}$ $a(6)=1$, tal que:

$$D_{1,1} = D_{2,2}$$

$$0 = 0$$

Neste caso particular em que $D_{8,2}=1$ acontece que $D_{5,1}=1$ e $D_{4,1}=1$, significando que ambas satisfazem a igualdade

$$D_{a(k+1),i(k+1)} = D_{a(k),i(k)} \text{ e } i(k+1) = i(k)-1.$$

Como o algoritmo não faz seleção, qualquer um deles pode pertencer ao caminho, por isso o caminho encontrado não é o caminho mais curto cujo comprimento é 7.

3.4.1. Observações Relativas à Segunda Parte do Algoritmo

Primeira Observação

Wollmer quando apresenta o algoritmo para determinação do caminho mais curto depois de obtido seu valor, escolhe para reconstituí-lo, aqueles nós em que:

$$D_{a(k+1), i(k+1)} = D_{a(k), i(k)} - \ell[a(k), a(k+1)] \text{ e } i(k+1) = i(k) \\ \text{ou } D_{a(k+1), i(k+1)} = D_{a(k), i(k)} \text{ e } i(k+1) = i(k) - 1.$$

É verdade que todos os nós pertencentes ao caminho mais curto satisfazem a pelo menos uma destas condições. Mas podem existir nós para cujos valores estas condições são verificadas e que não pertencem ao caminho desejado. Para ilustrar basta verificar o exemplo 6.

Segunda Observação

Depois de obtido o caminho mais curto, o algoritmo apresenta os arcos $[a(k), a(k+1)]$ onde $i(k) \neq i(k+1)$ como aqueles que falharam. Na realidade estes arcos são aqueles que falharam na determinação de $D_{a,k}$ ($\forall a$, não necessariamente $a=t$). Ocorre que um arco que falha no caminho mais curto do nó s até o nó $a \neq t$, pode ser um que não falhe no caminho mais curto do nó s até o nó t , como pode ser visto no exemplo 5.

Após estas observações terem sido feitas, procurou-se desenvolver um algoritmo para determinação do caminho mais curto e dos arcos que falharam utilizando o próprio processo de determinação do valor do caminho mais curto. Assim é o algoritmo proposto - CAMI.

C A P Í T U L O I V

4. ALGORITMO PROPOSTO - CAMI

A análise do algoritmo de Wollmer mostrou que alguns problemas podem surgir quando de sua aplicação. Para evitar estas situações e resolver o problema com um pequeno tempo de execução, as modificações já apresentadas no item 3.2. e algumas outras foram incorporadas ao programa, resultando o algoritmo CAMI.

Procedimento para determinação do caminho mais curto, seu valor e os arcos que sofreram redução:

```

INICIO
  i←0
  Para todo nó a≠s faça  $D_{a,i} \leftarrow \infty$ .
   $D_{s,i} \leftarrow 0$ 
  k←0
  Enquanto k<n faça
    Início
      alteração←1
      Se k=0 então
        Início
          EXECUTE BLOCO KZERO.
          ind←1
           $ME_{ind} \leftarrow t$ 
          EXECUTE BLOCO CAMINHO1
        Fim
      Senão
        EXECUTE BLOCO KMAIORZERO.
        Para todo nó a faça  $nulo_a \leftarrow 0$ .
        kk←k
        ind←1
         $ME_{ind} \leftarrow t$ 
        EXECUTE BLOCO CAMINHO2.
        k←k+1
      Se k<n então
        EXECUTE BLOCO REINICIALIZAÇÃO.
    Fim
  FIM.
  
```

BLOCO KZERO

Enquanto alteração = 1 faça

 Início

 alteração ← 0

 Para cada arco (a,b)

 Se $D_{b,k} > D_{a,k} + \ell(a,b)$ então

 Início

$D_{b,k} \leftarrow D_{a,k} + \ell(a,b)$

$LE_{b,k} \leftarrow a$

 Fim

 Senão

 Se $D_{a,k} > D_{b,k} + \ell(a,b)$ e $b < t$ então

 Início

$D_{a,k} > D_{b,k} + \ell(a,b)$

$LE_{a,k} \leftarrow b$

 alteração ← 1

 Fim

 Fim

FIM BLOCO KZERO.

BLOCO CAMINHO1

Enquanto $ME_{ind} > 1$ faça

 Início

$c \leftarrow ME_{ind}$

$ME_{ind+1} \leftarrow LE_{a,k}$

$ind \leftarrow ind+1$

 Fim

FIM BLOCO CAMINHO1.

BLOCO KMAIOR ZERO

Enquanto alteração = 1 faça

Início

alteração \leftarrow 0

Para cada arco (a,b)

Se $D_{b,k} > D_{a,k-1}$ então

Início

$D_{b,k} \leftarrow D_{a,k-1}$

$LE_{b,k} \leftarrow a$

$falha_{b,k} \leftarrow 1$

Fim

Senão

Se $D_{a,k} > D_{b,k-1}$ então

Início

$D_{a,k} \leftarrow D_{b,k-1}$

$LE_{a,k} \leftarrow b$

$falha_{a,k} \leftarrow 1$

Fim

Se $D_{b,k} > D_{a,k} + \ell(a,b)$ então

Início

$D_{b,k} \leftarrow D_{a,k} + \ell(a,b)$

$LE_{b,k} \leftarrow a$

$falha_{b,k} \leftarrow 0$

Fim

Senão

Se $D_{a,k} > D_{b,k} + \ell(a,b)$ e $b < t$ então

Início

$D_{a,k} \leftarrow D_{b,k} + \ell(a,b)$

$LE_{a,k} \leftarrow b$

$falha_{a,k} \leftarrow 0$

alteração \leftarrow 1

Fim

Fim

FIM BLOCO KMAIOR ZERO.

BLOCO CAMINHO2

Enquanto $ME_{ind} > 1$ faça

 Início

$c \leftarrow ME_{ind}$

$ME_{ind+1} \leftarrow LE_{c, kk}$

 Se $falha_{c, kk} = 1$ faça

 Início

$Nulo_{ind} \leftarrow ME_{ind+1}$

$kk \leftarrow kk-1$

 Fim

$ind \leftarrow ind+1$

 Fim

FIM BLOCO CAMINHO2.

BLOCO REINICIALIZAÇÃO

Para todo $n\tilde{o}$ a faça

 Início

$D_{a, k} \leftarrow D_{a, k-1}$

$LE_{a, k} \leftarrow LE_{a, k-1}$

 Fim

FIM BLOCO REINICIALIZAÇÃO.

O caminho mais curto é dado pelos nós ME_{ind} .

O valor deste caminho é $D_{t, n}$.

Os arcos que falharam foram aqueles ME_{q+1} , ME_q tais que $nulo_q \neq 0$.

4.1. Fluxograma do Algoritmo CAMI para Obtenção do Valor do Caminho mais Curto

Para melhor mostrar as modificações que foram feitas no algoritmo de Wollmer, é apresentado a seguir o fluxograma do algoritmo CAMI, apenas para obtenção do valor do caminho mais curto.

4.2. Exemplo de CAMI

Para mostrar o resultado da aplicação do algoritmo CAMI, este foi aplicado a uma rede de 50 nós e 129 arcos, cuja saída do computador aparece no exemplo 7.

Exemplo 7: Para a rede da Figura 12 (Figura 8c), o algoritmo CAMI fornece:

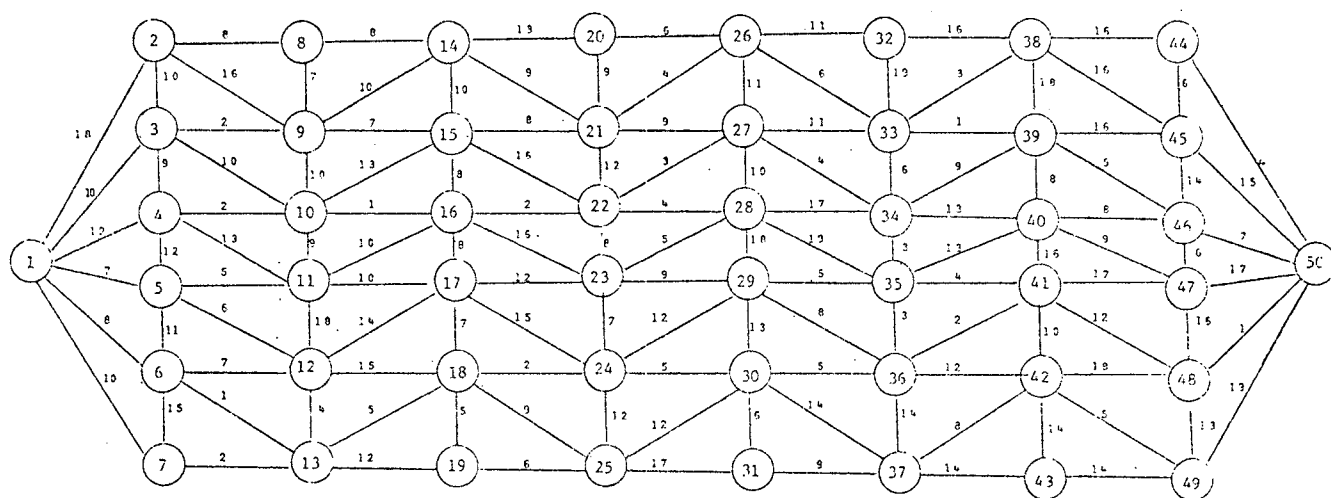


FIGURA 12 - Rede dual do exemplo 7.

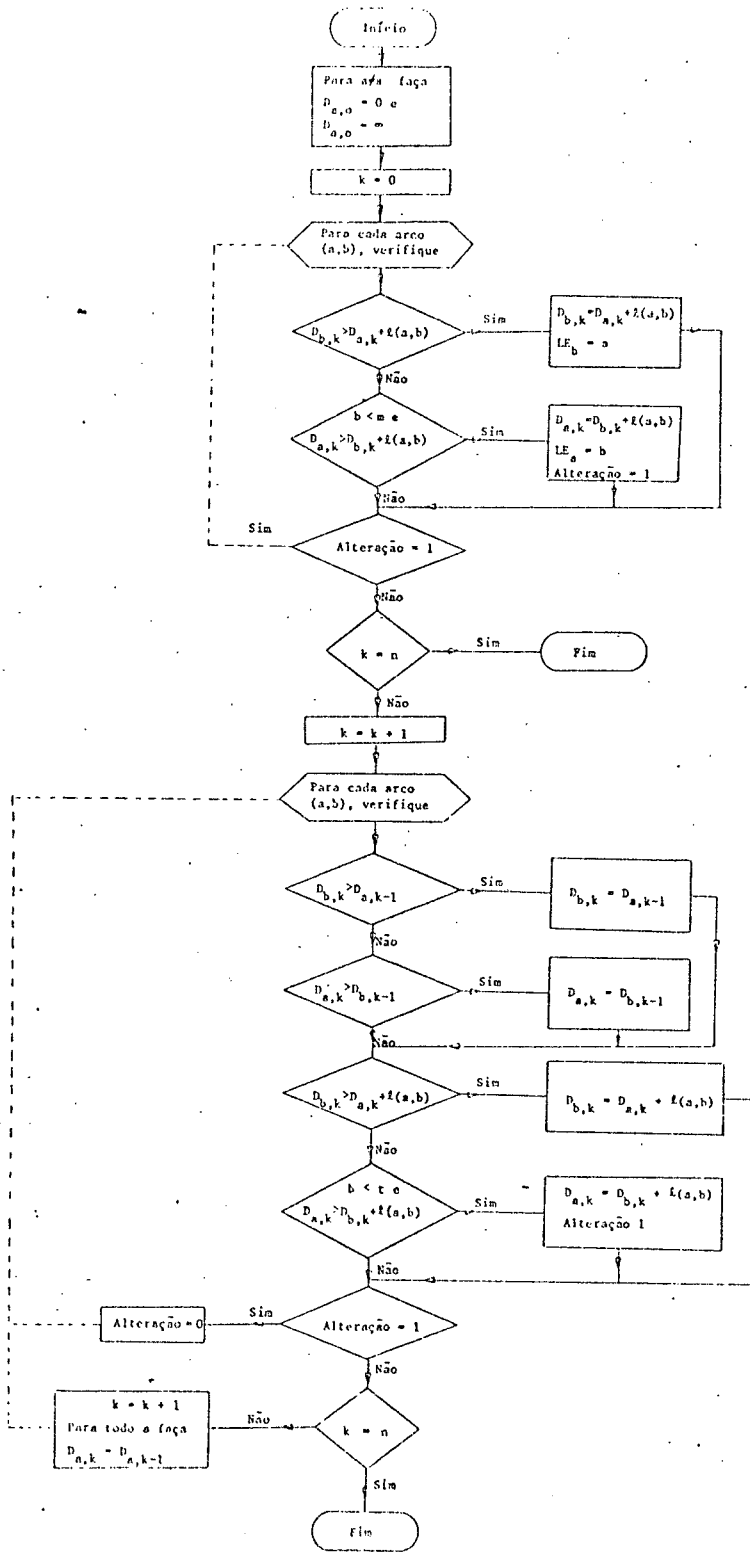


FIGURA 11 - Fluxograma de CAMI para obter o valor do caminho.

M E O NUMERO DE NOS REDE = 50
 N E O NUMERO DE ARCOS A SEREM REDUZIDOS = 4
 NV E O NUMERO DE ARCOS DA REDE=129

O VALOR DESEJADO E D(50,0)= 41.00

O CAMINHO MINIMO PASSA PELOS NOS

50 48 41 36 30 24 18 13 6 1

O VALOR DESEJADO E D(50,1)= 29.00

O CAMINHO MINIMO PASSA PELOS NOS

50 48 41 36 30 24 18 13 6 1

UM ARCO QUE FALHOU FOI 41 48 COM VALOR 12.00

O VALOR DESEJADO E D(50,2)= 20.00

O CAMINHO MINIMO PASSA PELOS NOS

50 48 41 36 34 27 22 16 10 4 1

UM ARCO QUE FALHOU FOI 41 48 COM VALOR 12.00

UM ARCO QUE FALHOU FOI 1 4 COM VALOR 12.00

O VALOR DESEJADO E D(50,3)= 14.00

O CAMINHO MINIMO PASSA PELOS NOS

50 46 39 34 27 22 16 10 4 1

UM ARCO QUE FALHOU FOI 46 50 COM VALOR 7.00

UM ARCO QUE FALHOU FOI 27 39 COM VALOR 11.00

UM ARCO QUE FALHOU FOI 1 4 COM VALOR 17.00

O VALOR DESEJADO E D(50,4)= 9.00

C A P Í T U L O V

5. SUGESTÕES PARA NOVAS PESQUISAS

Uma futura pesquisa pode ser feita de forma a considerar reduções parciais nos fluxos dos arcos. Neste caso pode-se estipular quais serão as reduções permitidas ou então fazer uma análise de sensibilidade considerando várias resoluções.

Outro estudo pode ser desenvolvido considerando que os arcos sejam funções e não valores fixos. Dessa forma, o algoritmo apresentado aqui pode também ser aplicado para que alternativas de investimentos sejam obtidas. Neste caso, cada alternativa de investimento, representada por uma rede, constituirá um nó da rede de decisão.

B I B L I O G R A F I A

1. BOAVENTURA NETTO, P.O. Teoria e modelos de grafos. São Paulo, Edgar Blücher, 1979.
2. DOULLIEZ, P.J. & RAO, M.R. Capacity of a network with increasing demands and arcs subject to failure. Operations research, 19(4):905-15, jul./aug. 1971.
3. DREYFUS, S.E. An Appraisal of some shortest-path algorithms. Operations Research, 17(3):395-412, 1969.
4. FORD, L.K. & FULKERSON, D.R. Flows in networks. Princeton, 1974.
5. FURTADO, A.L. Teoria dos Grafos - Algoritmos. Livros Técnicos e Científicos, 1973.
6. HU, T.C. Integer programming and network flows. U.S.A., Addison-Wesley, 1970.
7. SAKAROVITH, M. Optimisation dans les réseaux. Techniques Mathématiques de la Recherche Operationnelle. Grenoble, 1977.
8. WOLLMER, R. Removing arcs from a network. Operations Research, 12:934-40, 1964.